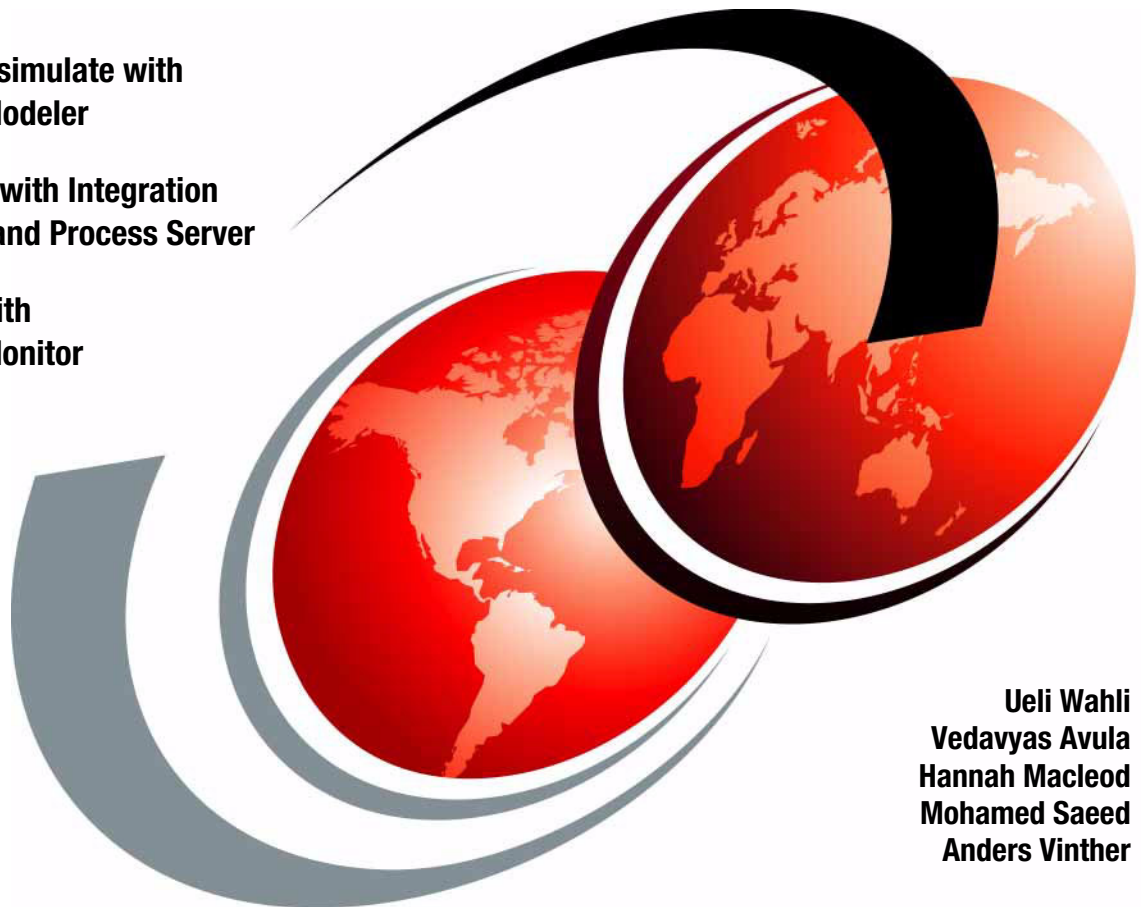


Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products

Model and simulate with
Business Modeler

Implement with Integration
Developer and Process Server

Measure with
Business Monitor



Ueli Wahli
Vedavyas Avula
Hannah Macleod
Mohamed Saeed
Anders Vinther



International Technical Support Organization

**Business Process Management:
Modeling through Monitoring
Using WebSphere V6.0.2 Products**

August 2007

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

Second Edition (August 2007)

This edition applies to WebSphere Business Modeler Version 6.0.2, WebSphere Integration Developer Version 6.0.2, WebSphere Process Server Version 6.0.2, and WebSphere Business Monitor Version 6.0.2.

© Copyright International Business Machines Corporation 2006, 2007. 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	xvii
Trademarks	xviii
Preface	xix
The team that wrote this IBM Redbooks publication	xx
Become a published author	xxii
Comments welcome	xxii
Summary of changes	xxiii
August 2007, Second Edition	xxiii
Changes in the business model (WebSphere Business Modeler)	xxiii
Changes in the implementation (WebSphere Integration Developer)	xxiv
Addition of the Monitor Toolkit (WebSphere Integration Developer)	xxiv
Changes in deployment (WebSphere Process Server)	xxiv
Changes in monitoring (WebSphere Business Monitor)	xxiv
Part 1. Introduction	1
Chapter 1. Business process management	3
What is a business process?	4
What is business process management?	4
Benefits of BPM	5
Information about business process management	5
IBM business process management solution	6
Business performance management	7
IBM business process management products	8
Service-oriented architecture (SOA)	9
Benefits of SOA	9
IBM SOA foundation	10
IBM BPM solution on the SOA foundation	11
Service Component Architecture	11
Service Data Objects	12
Information about SCA and SDO	13
More on business performance management	13
Summary	14
Chapter 2. Product overview	15
WebSphere Business Modeler	16
Complete and robust process modeling tool	17

Revolutionizes business flexibility	18
WebSphere Business Modeler Version 6.0.2.	19
New Business Measures view	19
Process modeling enhancements	19
Simulation and dynamic analysis enrichment	20
Reporting and printing improvements	20
Import and export enhancements	21
Interoperability improvements	21
Publishing enhancements	21
WebSphere Business Modeler Publishing Server	21
WebSphere Integration Developer	22
Eclipse-based development tool	22
SOA and SCA	23
Integration with other development tools	23
Business process support	23
WebSphere Integration Developer Version 6.0.2.	25
Visual snippets	25
Rational RequisitePro Integration	25
Top-down Web services development	25
Business rules	25
Business state machines	26
Human tasks	26
Information service	26
Adapters	26
Monitor Development Toolkit	27
Event definition editor	27
Refactoring	27
WebSphere Process Server	28
SOA core	28
Supporting services	30
Service components	32
WebSphere Process Server version 6.0.2	33
Administration	33
Business Process Choreographer Explorer	34
Business Process Choreographer Observer	34
Interoperability	34
WebSphere Business Monitor	34
WebSphere Business Monitor Version 6.0.2	36
Monitor Development Toolkit	36
Summary	37
Chapter 3. WebSphere Business Monitor Version 6.0.2 architecture	39
Overview	40

Overview of the Modeler to Monitor closed loop cycle	41
WebSphere Business Monitor scenarios	43
WebSphere Business Monitor architecture	44
Monitor Development Toolkit	46
Monitor Model Editor	46
Monitor model terminology	47
Unit test environment	51
WebSphere Business Monitor components	52
Monitor Server.	52
Monitor databases.	54
Monitor Dashboard	57
WebSphere Business Monitor topology	61
Simple Monitor topology	63
Single system topology	64
Summary	64
Chapter 4. Case study: ClipsAndTacks	65
Case study: ClipsAndTacks Office Supplies Ltd.	66
Background	66
Business problem	66
Business objectives.	67
Current order handling process.	68
Planned revisions to the order handling process	68
Key performance indicators	69
Roles	70
Subsystems	71
Hardware prerequisites	73
Summary	74
Part 2. Modeling the business	75
Chapter 5. Modeling the current business process	77
Introduction to modeling the current business process	78
Documenting the current process	78
Process flow	79
Business items	80
Resources	81
Timetables.	81
Creating the current process diagram using the Modeler	81
Detailed steps to create the model	81
Modeler quick guide	82
Importing the current process model using the Modeler	83
Order Handling (Current) process.	86
Business items	88

Resources	90
Organizations and classifiers	90
Summary	90
Chapter 6. Simulating and analyzing the current process	91
Overview of process simulation	92
ClipsAndTacks process assessment for the simulation	92
Role resources matrix	92
Duration matrix	93
Availability matrix	94
Decision probabilities	94
Simulation profile information	95
Populating the simulation environment	96
Populating role resource information in the project	96
Populating duration information in the process	99
Validating the simulation data	101
Populating probability information in the decision nodes	102
Populating probability information in activity nodes	102
Simulating the current process	103
Process instance simulation	103
Creating a simulation snapshot	103
Running the simulation	109
Analyzing the simulation results (current)	111
Process duration	112
Process cases summary	114
Resource usage	116
Process cost	118
Closing the reports	119
Other reports	119
More information	119
Summary	120
Chapter 7. Modeling the Future 1 business process	121
Documenting the Future 1 process	122
Business revision	122
Technical outline of the implementation in the Modeler	124
Building the Future 1 process	124
Creating timetables	124
Creating a new non-consumable bulk resource	127
Creating the new process: Order Handling (Future 1)	129
Overview of the new process	130
Deleting an activity	131
Creating new activities	131

Creating new decisions	132
Creating merge nodes	134
Creating connections	134
Asynchronous process	138
Changing the business item of the process input	138
Populating resources for the activities	138
Populating expression in decisions	139
Populating business comments in the process	142
Validating the process	143
Organizing the diagram	144
Building an overall process for simulation	145
Creating the main process Overall Order (Future 1)	146
Business Process Execution Language	148
Preparing the process for export	148
Defining the type of implementation for all activities	150
Defining the operation type for each activity	156
Defining the operation type for the process	156
Exporting the BPEL for WebSphere Integration Developer	157
Summary	162
Chapter 8. Simulating and analyzing the Future 1 process	163
Overview of simulating the Future 1 process	164
ClipsAndTacks process assessment for the simulation	164
Roles and costs matrix	165
Duration matrix	166
Availability matrix	167
Probabilities on process decisions	167
Populating the simulation environment	168
Entering simulation attributes in the Modeler	168
Populating role resources with costs and availability	168
Populating activity duration	169
Populating probabilities on decision output branches	171
Simulating the Future 1 process	172
Process instance simulation	172
Running the simulation	177
Analyzing the Future 1 simulation results	177
Process duration	178
Process cases summary	179
Resource usage	181
Process cost	181
Process comparison analysis	182
Processes duration comparison	183
Processes cost comparison analysis	184

More information	184
Additional considerations	184
Perspective	185
Summary	185
Chapter 9. Defining KPIs and measures	187
Business measures	188
Introduction to business measures	188
Model, assemble, deploy, and manage the complete life cycle	188
Elements of the Business Measures editor.	190
Monitor Development Toolkit	192
WebSphere Business Monitor dashboard views.	192
ClipsAndTacks key performance indicators	193
Definition of ClipsAndTacks KPIs and measures	193
Creating the business measures model.	194
Preparation	194
Monitored values.	196
Defining the KPIs	197
Average Process Duration	197
Percentage of Orders Shipped	200
Define the measures	202
Creating the Order Count measure.	202
Creating the other measures.	204
Dimensional analysis	204
Where do we define the details?	205
Exporting the business measures	205
Exporting the process model for execution.	207
Summary	208
Part 3. Development and testing	209
Chapter 10. Developing the application using WebSphere Integration Developer	211
Overview of the application implementation.	212
Setting up a database to keep the customer orders.	212
Description of the database tables	213
Populating the database	214
Physical database.	214
Configuring the Process Server test environment	215
Starting Integration Developer	215
Starting the process server and administrative console	216
Creating a data source for the database.	218
Importing the model	222
Importing the project interchange file	222

Business Integration view	224
Assembly diagram	225
Process diagram	226
Business objects (data types)	227
Physical resources	228
Changing the namespace	229
Implementing a business rule	229
Implementing the business rule group	230
Defining a business rule based on a template	231
Implementing the Java activities	233
Check Customer Account Status	233
Cancel Order and Send Notification	235
Update Order Database	237
Configuring the human tasks	237
Testing the application	239
Deployment of the application for testing	239
Using the built-in process test facility	239
Processing the human task using the BPC Explorer	241
Rerunning tests and reusing data	245
Adding a stand-alone reference to invoke the process	245
Implementing a Web front-end	247
Architecture	247
Importing the Java module	251
Importing the Web application	252
Attaching the Web front-end to the process application	254
Initializing the database	255
Listing the database content	256
Implementing customized JSPs for the BPC Explorer	257
Importing the human task JSPs	258
Configuring the BPC Explorer for customized JSPs	259
Human task JSP processing logic	260
Running the application in the test environment	262
Using the Web front-end	263
Using the human task JSPs with the BPC Explorer	265
Implementing an external Web service	267
Importing the Web service	267
Adding the Web service to the assembly diagram	268
Invoking the Web service from an activity	269
Testing the application with the Web service	270
Changing the port for the Web service	271
Generating a human task application	273
Improving the generated user interface	277
Implementing a customized human task application	279

Importing the human task application	279
Architecture	280
Using the human task application	284
Implementing database update	287
Using an information service	287
Using JDBC in a Java activity	292
Using the Business Rules Manager	295
Using the Business Rules Manager	295
Exporting business rules	297
Using the Debugger	298
Setting breakpoints	298
Debugging the application	299
Summary	302
Chapter 11. Implementing security	303
Implementing security for human tasks	304
Using a custom user registry for security	304
Defining groups and users	304
Specifying security in the Process Server	306
Configuring security roles for submitting events	306
Using the groups for human tasks	312
Testing human task security with the BPC Explorer	314
Using the human task Web application with security	314
Using security in the generated user interface application	318
Using an LDAP server	318
Groups and users	318
Configuring the Process Server for LDAP	318
Configuring the human tasks for LDAP	324
Testing LDAP security	324
Removing security	325
Summary	325
Chapter 12. Developing and testing the business measures with the Monitor Toolkit	327
Preparation in Modeler	328
Monitor Toolkit	328
Importing business measures into the Monitor Toolkit	328
Studying the imported business measures	330
Creating the monitor model	333
Generating CEI events for BPEL elements	334
Generating monitoring events	337
Generating the monitor model	338
Monitor Model Editor	339

Creating metrics and measures	341
Order Count	341
Shipped Order Count	345
Declined Order Count	346
Order Price Total	347
Order Price Average	349
Creating a dimension	349
Location dimensions by country and city	350
Creating KPIs	353
Average order fulfillment time for shipped orders	353
Percentage of orders shipped	359
Creating alerts	362
Shipped order fulfillment time is greater than 3 days and 1 hour	362
Percentage of shipped orders are less than 85%	367
Implementing round-trip measures	369
Determining activity duration	369
Determining branch percentages	371
Implementing round-trip measures for human tasks	372
Creating the monitor models for human tasks	372
Merging the monitor models	373
Business measures for human task processing time	377
Displaying business metrics and KPIs in the visual model	379
Modeler feedback loop	379
Monitor model content	383
Deploying the monitor model to the unit test environment	384
Configuring the Monitor Server	384
Deploying the business process application	385
Deploying the monitor model application	385
Defining the data source in the Monitor Server	391
Testing the monitor model	392
Monitor Dashboard in the unit test environment	393
Exporting the monitor model EAR file	396
Undeploying the monitor model in the test environment	396
Summary	397

Part 4. Deployment and monitoring 399

Chapter 13. Deploying and running the application in Process Server	401
Exporting the application from Integration Developer	402
Preparation	402
Exporting the EAR files	402
Configuring the Process Server	403
Using a JACL script to define the data source	404

Configure security in Process Server	404
Installing the application in Process Server	405
Running the application (without monitor)	406
Creating the database	406
Running the Web front-end to submit an order	406
Using the BPC Explorer	406
Using the customized human task application	407
Using the Business Rules Manager	407
Accessing the server log	408
Using the Common Base Event Browser	408
Uninstalling the application	412
Summary	413
Chapter 14. Deploying the monitor model and measuring the Future 1 process	415
Deploying the monitor model	416
Installing the monitor model	416
Installing the monitor model application	416
Making the application startable	419
Configuring the Adaptive Action Manager	441
Connecting the Action Manager to LDAP	441
Defining the notification templates	443
Binding the notification templates to situation events	446
DB2 replication scripts	449
Monitor model undeployment	451
Stopping the monitor model	451
Removing the Alphablox cubes	452
Removing the DB2 Cube Views configuration	453
Removing the DMS configuration	453
Removing the model schema	454
Uninstalling the monitor model version application	455
Troubleshooting	456
What next?	457
WebSphere Business Monitor Dashboard	458
Copying the WebSphere Business Monitor portlets	458
Creating a portal page for ClipsAndTacks	461
Configuring the dashboard views	465
Authorization	475
Measuring the Future 1 application	477
Tracking the orders in the Monitor databases	478
Analyzing the Future 1 dashboard data	480
Instances view analysis	480
Key performance indicator view analysis	481

Gauge view analysis	481
Alerts view analysis	481
Dimensional view analysis	483
Other views, and wiring views together	484
Database replication problem	484
Summary	485
Part 5. Continuous process improvement	487
Chapter 15. Comparing the measurements and modeling the Future 2 process	489
Exporting data from the Monitor	490
Importing data into the Modeler	491
Analyzing the Monitor measurements	494
Comparing the measurements with the simulation	495
KPI analysis	495
Developing a strategy for improvements	495
Automatically approving more orders	495
Automatically declining orders for customers with low credit rating	496
Creating the Future 2 process in the Modeler	496
Changing the account status check to a global process	499
Order Handling (Future 2)	500
Modeling the global subprocess	502
Importing the Web service	502
Creating the process flow	503
Setting the attributes	504
Deleting the overall process	505
Simulating the Future 2 process	505
Analyzing the Future 2 simulation results	505
Future 2 business measures	506
Exporting for Integration Developer and Process Server	506
Summary	507
Chapter 16. Implementing the Future 2 process using WebSphere Integration Developer	509
Importing the Future 2 application	510
Assembly diagram	511
Process diagram	511
Completing the application in Integration Developer	513
Updating assembly diagram	513
Implementing the business rule	515
Implementing the human tasks	518
Adding the Update Order information service for a shipped order	520
Implementing the Cancel Order and Send Notification component	520

Adding an information service for a declined order	521
Implementing the Check Customer Credit component	522
Implementing the Update order with Customer Credit component	525
Updating the customer information in the database	526
Importing the custom human task application	527
Importing the Web front-end application	528
Deploying and testing the Future 2 application	533
Testing the application	533
Changing the port numbers	534
Implementing a new Web service	536
Importing the new Web service	536
Updating the business process	537
Deploying and testing the application with the new Web service	541
Implementing a fault handler	541
Installing and running the application in Process Server	543
Exporting the Future 2 application	543
Installing the application	543
Running the application	543
Testing the business rule	544
Summary	544
Chapter 17. Measuring the Future 2 process	545
Creating the monitor model for Future 2	546
Preparing the project and the model	546
Merging the models	546
Simplified approach	546
Creating the metrics and the measures	547
Creating dimensions	549
Creating KPIs	550
Creating alerts	550
Measuring duration times and decision percentages	551
Monitoring model content	551
Deploying the model to the Monitor test environment	552
Testing the Future 2 monitor model	552
Deploying the Future 2 monitor model	554
Measuring the Future 2 application	555
Monitor Dashboard for the Future 2 application	555
Gauge view	555
KPI view	556
Instances view	556
Dimensions view	556
Alerts view	560
Reports view	561

Wiring two portlets on one portal page	562
Using the visual model	564
Displaying instances in the visual model	564
Displaying KPIs in the visual model	566
Dashboard portal page to display the visual model	566
Installing a new version of the monitor model	569
Steps to deploy a monitor model version	570
Summary	571
Part 6. Appendices	573
Appendix A. Installation of the development products	575
Installation of WebSphere Business Modeler V6.0.2	576
Interim fixes and fix packs for the Modeler	577
Starting the Modeler	577
Installation of WebSphere Integration Developer V6.0.2	578
Interim fixes and fix packs for Integration Developer	582
Interim fixes and fix packs for the process server test environment	582
Installing the information service feature	582
Starting WebSphere Integration Developer	583
Installation of the Monitor Development Toolkit	584
Changing the port number	587
Installation of WebSphere Process Server V6.0.2	588
Interim fixes and fix packs for Process Server	596
Controlling the Process Server	597
Creating the database for the business process container	598
Configuring the business process container	600
Configuring the human task container	603
Enabling security for the Process Server	603
Configuring security roles for submitting events	608
Verifying the installation	610
Installation of IBM Tivoli Directory Server V5.2	612
Directory server configuration	614
Starting the LDAP server	616
Groups and users	617
Appendix B. Installation of WebSphere Business Monitor	619
WebSphere Business Monitor prerequisites	620
System requirements	620
Software requirements	620
Installation directories	621
Prerequisites	622
Installing the Monitor Server and the Dashboard Server	622
Installing the Monitor Server (Process Server)	625

Installing the Dashboard Server	627
Starting the Monitor Server	631
Starting the Dashboard Server	631
Monitor Server service integration bus	632
SIB configuration for the Monitor Server	632
SIB configuration for the Process Server	634
Service integration bus post-configuration checkpoint	635
Implementing security for the Monitor	636
Securing the Monitor Dashboard	636
Securing the Monitor Server	639
Import LTPA keys	640
DB2 AlphaBlox	642
Fix pack installation	644
Appendix C. Additional material	647
Locating the Web material	647
Using the Web material	648
System requirements for using the Web material	648
Memory requirements by product	648
How to use the Web material	649
Creating the CLIPTACK database	650
Importing Modeler solutions	650
Importing application development solutions	650
Importing application execution solutions	651
Importing application monitoring solutions	651
Abbreviations and acronyms	653
Related publications	655
IBM Redbooks	655
Online resources	656
How to get IBM Redbooks	657
Help from IBM	657
Index	659

Notices

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

AIX®
BetaWorks™
CICS®
ClearCase®
Cloudscape™

Cube Views™
DB2®
IBM®
IMS™
Rational®

Redbooks®
Redbooks (logo) ®
RequisitePro®
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

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

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

Snapshot, and the Network Appliance logo are trademarks or registered trademarks of Network Appliance, Inc. in the U.S. and other countries.

EJB, Java, Java Naming and Directory Interface, JavaScript, JavaServer, JavaServer Pages, JDBC, JSP, J2EE, J2SE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

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

Preface

This IBM® Redbooks® publication presents a business process management (BPM) “improvement cycle” scenario, showing how a business can use a full business integration solution to complete the following tasks:

- ▶ Model and simulate a business process
- ▶ Develop and test an application to implement the business process
- ▶ Deploy and run the application on a server
- ▶ Implement and test business measures
- ▶ Monitor the application to observe pre-determined key performance indicators
- ▶ Import the observed data to make revisions to the original process model

The business scenario described in this document has been simplified to provide a full description of each stage of the BPM end-to-end process. To avoid having an overly large and unwieldy document, the authors focus on specific tasks, elements, and details, and not on presenting all possible facets of a complex business process.

This book is structured into five parts:

1. Business process management—This part provides a general introduction of the subject, the products involved, and an overview of the scenario that is used throughout the document.
2. Modeling the business—This part describes how to use **WebSphere® Business Modeler** to model the business process and its resources, simulate the process, and analyze the simulation results. Finally, it introduces the business measures defined for monitoring the process.
3. Development and testing—This part describes how to implement the business process application, exported from the Modeler, using **WebSphere Integration Developer** and Process Server test environment. The part also covers development and testing of the business measures using the **Monitor Development Toolkit** and the Monitor Server test environment.
4. Deployment and monitoring—This part describes how to deploy the application to **WebSphere Process Server** and how to deploy the business measures to **WebSphere Business Monitor** for monitoring the running application.
5. Continuous process improvement—This part provides information about the improvement cycle scenario of exporting monitor results back into Business Modeler, and improving, regenerating, and redeploying the modified application for further measurements.

The team that wrote this IBM Redbooks publication

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



Ueli Wahli is a Consultant IT Specialist at the IBM International Technical Support Organization in San Jose, California. Before joining the ITSO over 20 years ago, Ueli worked in technical support at IBM Switzerland. He writes extensively and teaches IBM classes worldwide about WebSphere Application Server, and WebSphere and Rational® application development products. In his ITSO career, Ueli has produced more than 40 IBM Redbooks. Ueli holds a degree in Mathematics from the Swiss Federal Institute of Technology.

Vedavyas Avula is a consultant at Miracle Software Systems. He graduated from University of Madras. He specializes in providing BPM and SOA solutions to the customers using IBM Products. His areas of expertise include WebSphere Business Modeler, Portal Server, Process Server, Business Monitor, Partner Gateway and Data Interchange. He has strong expertise in healthcare, retail, insurance, and finance industry verticals.

Hannah Macleod is a WebSphere IT Specialist in the IBM BetaWorks™ organization, specializing in WebSphere Business Modeler and Monitor. Prior to this, she had four years of experience in CICS®. Her roles included CICS TS Level 3 Service and technical focal point for the CICS TS 3.1 Beta Program. She has written a range of materials covering CICS TS and the CICS Tooling products, including several TechNotes and the book *Migration Considerations for CICS Using CICS CM, CICS PA, and CICS IA*, SG24-7294. She joined the IBM Hursley Lab (UK) in 2002 after graduating from Oxford University with a MPhys (Physics).

Mohamed Saeed is a Business Process Management consultant in Software Services for WebSphere in IBM Egypt. He graduated from the Computer Engineering department of the Cairo University in Egypt, and directly joined IBM in the testing team, then moved to the development team after one year and stayed there for two years. Mohamed then joined the IBM SWAT support team that handles critical customer situations, and finally joined IBM Software Services for WebSphere. During these years Mohamed gained valuable experience in different areas of service-oriented architecture, especially in monitoring business processes and J2EE™ applications.

Anders Vinther is an Accredited Senior IT Specialist in Australia. He has 15 years of experience in the IT field. He joined IBM Software Services for WebSphere in Denmark in 2000 and relocated to Sydney in late 2003. In 2005 he joined the Software Group Technical Sales team in the WebSphere brand and is now located in Melbourne. Prior to joining IBM he worked in various fields of software development, primarily in the financial industry, and he worked as a freelance consultant in South Africa, United Kingdom, and Australia before joining IBM in Denmark. His areas of expertise include WebSphere Application Server, WebSphere Portal Server, WebSphere Process Server, WebSphere Integration Developer, and WebSphere Services Registry and Repository.

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

- ▶ **Larissa Leybovich, Eric Prevost, Russell Scher, Andre Venancio, Sascha Wiederkom,** and **Neil MacKinnon**, who wrote the first edition of the IBM Redbooks publication, *Business Process Management: Modeling through Monitoring Using WebSphere V6 Products*, SG24-7148-00.
- ▶ **Joe Pappas**, IBM Somers, for his help in displaying process instances and metrics in a Monitor Dashboard diagram.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbooks publication dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, business partners, and clients.

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 Redbooks form found at:

ibm.com/redbooks

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

redbooks@us.ibm.com

- ▶ Mail your comments to:

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

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-7148-01
for Business Process Management: Modeling through Monitoring Using
WebSphere V6.0.2 Products
as created or updated on August 20, 2007.

August 2007, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

Changes in the business model (WebSphere Business Modeler)

The basic business model was left intact, but some changes were made to explore new function in WebSphere Business Modeler:

- ▶ We remove the customer classification data item from the current and Future 1 model, and add it to the Future 2 model. This requires changing the namespace in Integration Developer to have business objects of the same name but different content.
- ▶ We change the cancel of an order to be implemented in Java™ instead of a human task. An automated activity can generate the e-mail to the customer.
- ▶ We add a Java activity to update the database after shipping an order to remove the database update from the human task.
- ▶ For the Future 2 model we import the Web service WSDL file into the model and create a subprocess for the customer credit check. This subprocess uses a map to prepare the parameter for the Web service call, invokes the Web service, and uses the result to update the business object.
- ▶ We define only a few simple measures and KPIs in the Modeler (the complete business measures model of V6.0.1 is not supported any more).

Changes in the implementation (WebSphere Integration Developer)

- ▶ We describe how to change the namespace of business objects, which is required if you want to run multiple applications with business objects of the same name but different content.
- ▶ We define the business rule using a ruleset with a template in Future 1, and using a decision table with a template in Future 2. We use the Business Rules Manager during testing.
- ▶ We implement cancel order using Java code.
- ▶ We use an information service (database activity) to update the database after shipping an order.
- ▶ We show how to generate and tailor a human task application.
- ▶ We show how to use a Visual Snippet to prepare the parameters for the Web service call.
- ▶ In Future 2 we show the usage of interface and business object maps to exchange the external Web service to a service with a different interface, without writing any code.
- ▶ We show how to invoke the business process as a Web service.
- ▶ We implement security in the test phase before deployment. We use both a simple custom user registry and LDAP with the Tivoli® Directory Server.

Addition of the Monitor Toolkit (WebSphere Integration Developer)

- ▶ We define the business measures in a monitor model using the Monitor Toolkit, which is a plug-in in Integration Developer. We show different techniques for defining triggers, metrics, measures, and KPIs.
- ▶ We deploy and test the monitor model in the Monitor Toolkit test environment.
- ▶ We create an enterprise application from the monitor model.

Changes in deployment (WebSphere Process Server)

- ▶ We deploy the application with security.

Changes in monitoring (WebSphere Business Monitor)

- ▶ We implement a simple setup where the Monitor Server and the Dashboard Server (WebSphere Portal) run on the same computer system.
- ▶ We describe in detail how to deploy a monitor model and how to configure the Monitor Dashboard.



Part 1

Introduction

In Part 1 we introduce the concepts of business process management, the IBM process integration products that support business process management, and the case study, a company named *ClipsAndTacks*, that we use throughout the rest of the book.

We also describe the architecture of the WebSphere Business Monitor product, which has changed considerably in Version 6.0.2.



Business process management

In this chapter we introduce the concepts and benefits of business process management (BPM) and service-oriented architecture (SOA) that enables business value through operational flexibility, responsiveness, and reuse.

We also introduce the Service Component Architecture (SCA) and Services Data Objects (SDO) as building stones for the SOA architecture.

What is a business process?

If you search the Web for a definition of *business process*, you find many definitions. All these definitions contain certain common elements:

- ▶ A business process is triggered by a business event.
- ▶ A business process has an input and creates an output that is of value to the organization, its stake holders, or its customers.
- ▶ A business process is composed of related structural activities. Material and/or information flows between the process activities.
- ▶ A business process can be part of a larger process and can include or depend on other business processes.
- ▶ A business process can be viewed as the workflow for a use case.
- ▶ A business process usually depends upon several business functions for support, for example, IT, personnel, and accommodation.

What is business process management?

Business process management (BPM) leads to business innovation and optimization by implementing business strategy through modeling, developing, deploying, and managing business processes throughout their entire life cycle. BPM acts as an enabler for the businesses in defining and implementing strategic business goals and then measuring and managing company's financial and operational performance against these goals. The power of optimal results from the BPM life cycle activities is derived from the integrated set of the robust technology infrastructure and tools.

Business process management provides a convergence of technology that removes business and IT constraints through integration and enhanced technology to help streamline business transformation. These capabilities provide tight integration of operational and analytical environments, business and IT environments, and strategy with daily operations.

Business process management combines business processes, information, and IT resources, aligning your 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. This integration of resources allows your organization to obtain business information faster, respond more quickly to market trends and competitive threats, and improve operational efficiencies and business results—all attributes of an on demand enterprise.

Benefits of BPM

BPM 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
- ▶ Reduce risk by gaining an understanding of process impacts prior to operationalizing
- ▶ 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

The robust business process management solution will provide powerful tools for the business and IT side of the business 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 the 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

Information about business process management

You can find definitions and more information about business process management on the Internet, for example:

- ▶ Wikipedia:
http://en.wikipedia.org/wiki/Business_Process_Management
- ▶ Business Process Management Initiative:
<http://www.bpmi.org/>

IBM business process management solution

The IBM process integration portfolio provides capabilities required for the delivery of the comprehensive enterprise wide business process management strategies and solution. 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 BPM solution includes development tools, used to implement custom artifacts that leverage the infrastructure capabilities, and business performance management tools, used to monitor and manage the runtime implementations at both the IT and business process levels.

Business process management allows companies to implement the continuous end-to-end business process life cycle in an open environment (Figure 1-1).

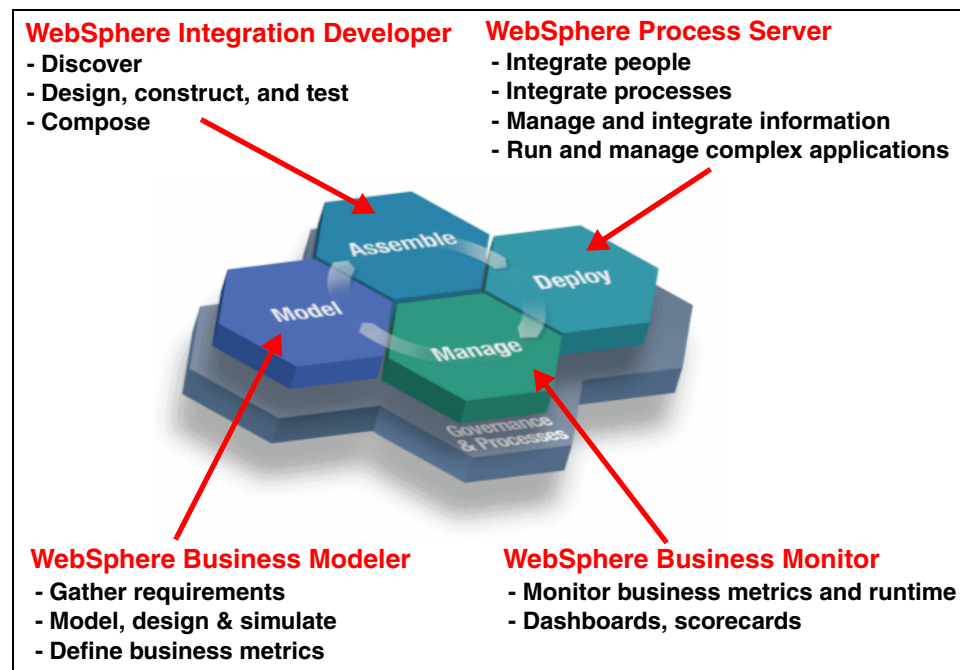


Figure 1-1 Business process management life cycle

The IBM BPM solution component tools support the following major activities:

- ▶ **Model**—Capture, simulate, analyze, and optimize business models to reduce risk and increase flexibility
- ▶ **Assemble**—Develop, assemble, and test integrated solution
- ▶ **Deploy**—Direct deployment of models and policies to realize business intent

- ▶ **Manage**—Manage the deployed models:
 - **Monitor** and correlate metrics and alerts in real-time from internal and external sources to gain visibility into the business and IT performance
 - **Analyze performance** results to gaining insight into the business metrics and information for contextual based decision making
 - **Act** by responding at the right time to insights through collaboration, optimization, and automation to excel

One key feature of the IBM business process management is the linkage between the **Manage** and the **Model** life cycle stages. This functionality enables the delivery of runtime data and statistics into the process modeling environment to allow for the completion of the analyses that drives iterative process re-engineering through a continuous business process improvement cycle.

Business performance management

Business performance management tools incorporate monitoring capabilities that aggregate operational and process metrics in order to efficiently manage systems and processes. Managing these systems requires a set of capabilities that span the needs of IT operations professionals and business analysts who manage the business operations of the enterprise.

These capabilities are delivered through a set of comprehensive services that collect and present both IT and process-level data, allowing business dashboards, administrative dashboards, and other IT-level displays to be used to manage system resources and business processes.

Through these displays and services, it is possible for line-of-business (LOB) and IT personnel to collaborate to determine, for example, what business process paths may not be performing at maximum efficiency, the impact of system problems on specific processes, or the relationship of system performance to business process performance.

This collaboration allows IT personnel and assets to be tied more directly to the business success of the enterprise than they traditionally have been.

For more information about business performance management, refer to:

- ▶ IBM business performance management community:
<http://www.ibm.com/software/info/topic/perform/partnerpage.html>
- ▶ Wikipedia:
http://en.wikipedia.org/wiki/Business_performance_management

IBM business process management products

The BPM solution consists of four separate products:

- ▶ WebSphere Business Modeler
- ▶ WebSphere Integration Developer
- ▶ WebSphere Process Server
- ▶ WebSphere Business Monitor

Together, these products enable businesses and other organizations to plan and implement a unified business process strategy based on realistic simulations and observed data.

At a high level, here is how you use these products together to get the right BPM solution for your business needs:

- ▶ With **Modeler** you to begin the cycle by designing the optimal business process for a particular case. Typically, Modeler is used by a business analyst to specify *what* activities have to be done and in which order. Modeler can export a business process as Business Process Execution Language (BPEL) for implementation.
- ▶ With **Integration Developer** you implement the business process model as an executable IT flow. This is done by mapping the model activities to reusable service components to construct composite business applications. Typically, Integration Developer is used by a developer to specify *how* activities get implemented.
- ▶ **Process Server** provides the production server to run and manage the application you create. Typically, a system administrator works with Process Server.
- ▶ **Monitor** provides real time performance monitoring of the application. The **Monitor Development Toolkit**, a plug-in in Integration Developer, is used to specify and test the business measures, before deploying the measures to a Monitor Server. Monitor is usually set up and administered by a system administrator.

Monitor is built on the concept of event monitoring of any system that produces events. The terminology used to describe this is called *Business Activity Monitoring (BAM)*.

Once built, improvements to the model and application are on-going.

For more information about the IBM business process management products, refer to:

<http://www.ibm.com/developerworks/websphere/zones/businessintegration/>

Service-oriented architecture (SOA)

Service-oriented architecture (SOA) is the IT model that enables business value through operational flexibility, responsiveness, and reuse. SOA is an application framework that takes everyday business applications, breaks them into individual business functions and processes, called services, and then inter-relates them through well-defined interfaces and contracts.

The interfaces are defined in a neutral manner that is independent of the hardware platform, the operating system, and the programming language in which the service is implemented. This allows services, built on a variety of such systems, to interact with each other in a uniform and universal manner.

SOA is an evolution as opposed to a revolution. The growing market momentum around SOA means that companies that compete with you are adopting SOA to gain a strategic advantage. SOA would be impossible without industry standards. While there have always been standards in IT, finally we are seeing a very broad industry support for the standards underpinning SOA. This ensures that the standards are real, meaningful, and are here to stay.

Benefits of SOA

Establishing a service-oriented architecture can help prepare both IT and business processes for rapid change. Even in the early stages of adopting an SOA, your organization will benefit from:

- ▶ Increase revenue—Create new routes to market and create new value from existing systems
- ▶ Provide a flexible business model—React to market changes more quickly
- ▶ Drive down cost—Eliminate duplicate systems, build once and leverage, and improve time to market
- ▶ Reduce risk and exposure—Improve visibility into business operations

The SOA approach can bridge the gap between what you want your business to accomplish and the infrastructure tools you need to get there:

- ▶ Decrease development and deployment cycle times by using pre-built, reusable services building blocks
- ▶ Integrate across the enterprise—even historically separate systems—and facilitate mergers and acquisitions of enterprises
- ▶ Reduce cycle times and costs by moving from manual to automated transactions

- ▶ Make it easier to do business with business partners by increasing your flexibility
- ▶ Bring adaptable, scalable solutions to complex business problems by using best practices, such as layering and loosely-coupled components

IBM SOA foundation

The IBM SOA foundation is an integrated, open set of software, best practices, and patterns that provides you with what you need to get you started with SOA. The SOA foundation provides full support for the SOA life cycle through an integrated set of tools and runtime components that allow you to leverage skills and investments across the common runtime, tooling, and management infrastructure.

The components are modular, allowing you to pick and choose the pieces you need to deliver an immediate impact while knowing that what you pick will work with pieces you add later on. In addition, the SOA foundation is scalable, allowing you to start small and grow as fast as your business requires. The SOA foundation provides extensive support for business and IT standards; facilitating greater interoperability and portability between applications. It can also help you to leverage SOA to extend the value of the applications and business processes that are running your business today.

The SOA reference architecture (Figure 1-2) is a way of looking at the set of services that go into building an SOA. These capabilities can be implemented on a build as-you-go basis, allowing capabilities and project level solutions to be easily added as new requirements are addressed over time.

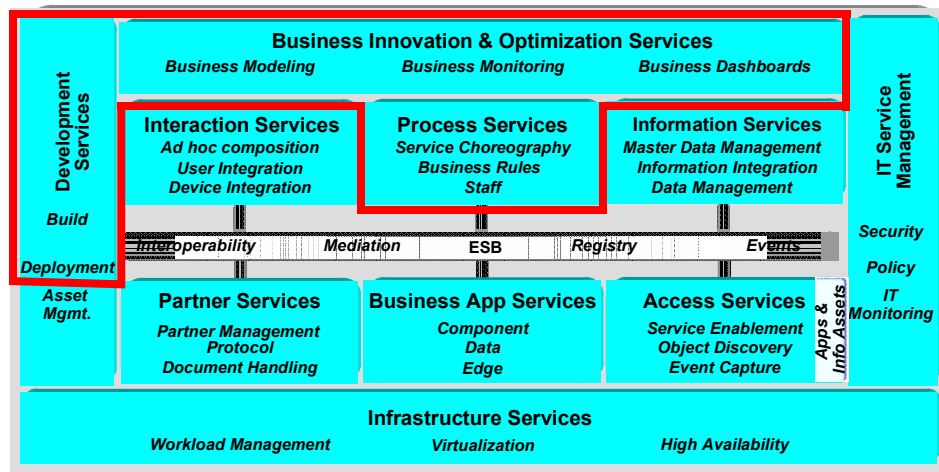


Figure 1-2 IBM SOA reference architecture

The backbone of the reference architecture is the enterprise service bus (ESB) that facilitates communication between services. The reference architecture is a great tool for laying out roadmaps for pursuing SOA.

Every component included in this architecture is provided by separate IBM products. The highlighted parts show the components that address the business process management area.

This service-oriented architecture provides a modular, scalable, portable and interoperable environment to support the equivalent aspects in the business area.

The reference architecture shows the tight integration with other critical IT aspects such as security, IT monitoring, virtualization, and workload management.

IBM BPM solution on the SOA foundation

The BPM solution available from IBM enables service-oriented, end-to-end process management for your organization based on SOA. Its SOA component applications enables businesses and organizations to plan, develop, implement, and improve their processes.

This solution lets organizations closely scrutinize costs, scheduling, resources, workflow, and other factors that affect the efficiency and viability of a given process, and determine the most effective methods for revising and improving that process.

Service Component Architecture

Service Component Architecture (SCA) is a fundamental enabler for service-oriented architecture (SOA), which structures IT assets as a series of reusable services that perform a business function.

SCA as a standard has now achieved very broad industry support with companies such as BEA, Oracle®, Red Hat Inc., SAP® AG, Sun Microsystems, Sybase, and TIBCO Software Inc., backing it in addition to IBM.

SCA is a technology to simplify application development and implementation in an SOA. With SCA, customers will be able to more easily create new and transform existing IT assets into reusable services that may be rapidly adapted to changing business requirements. Furthermore, these new technologies greatly reduce complexity associated with developing applications by providing a way to unify these services regardless of the programming languages they were written in and the platforms used to run them.

By structuring applications as a series of services, IT assets become more agile and organizations are better able to align their investments in dynamic business environments. In addition, adopting these new standards will provide organizations a higher degree of investment protection, as they will be able to reuse services with a variety of middleware technologies.

SCA provides an open, technology-neutral model for implementing IT services defined in terms of a business function, and do not unnecessarily expose the developer to the complexity of traditional middleware programming. SCA also provides a model for the assembly of business solutions from collections of individual services, with control over aspects of the solution such as access methods and security.

SCA gives developers and architects the ability to represent business logic as reusable components that can be easily integrated into any SCA-compliant application or solution. The resulting application is known as a composite application.

Service Data Objects

Service Data Objects (SDO) is a technology that was originally developed as a joint collaboration between BEA and IBM and is now being developed by BEA, IBM, Oracle, SAP, Sybase, XCalia and others. SDO is designed to simplify and unify the way in which applications handle data. Using SDO, application programmers can uniformly access and manipulate data from heterogeneous data sources, including relational databases, XML data sources, Web services and enterprise information systems.

SDOs specify a standard way to access data and can be used to modify business data regardless of how it is physically accessed. Developers and architects do not need to know the technical details of how to access a particular back-end data source in order to use SDO in their composite applications. Consequently, they can use static or dynamic programming styles and obtain connected as well as disconnected access.

SDO complements SCA by providing a common way to access many different kinds of data. The specification reduces the skill levels and time required to access and manipulate business data. Today, a multitude of APIs are used to manipulate data. These APIs tend to tightly couple the source and target of the data, making their use error-prone and subject to breaking as business requirements evolve. SDO makes it easier to use and realize the value of these APIs without having to code directly to them.

Information about SCA and SDO

In response to requests from customers and independent software vendor partners, IBM is jointly delivering two specifications for building systems that use a service-oriented architecture (SOA), which aim to provide developers with simpler and more powerful ways of constructing applications based on SOA: Service Component Architecture (SCA) and Service Data Objects (SDO).

These specifications are available at:

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

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

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

More on business performance management

There is no standardization of terms in this area. For example, Meta Group and Aberdeen use business performance management (BPM), Gartner refers to corporate performance management (CPM), and companies such as PeopleSoft® and Business Objects use the term enterprise performance management (EPM). IBM has adopted the business performance management term when referring to performance management practice and solutions.

As a first deliverable, the business performance management standards group has developed a common definition of business performance management that provides appropriate context for performance management, including the following principles:

- ▶ BPM is a set of integrated, closed-loop management and analytic processes, supported by technology, that address financial as well as operational activities.
- ▶ BPM is an enabler for businesses in defining strategic goals, and then measuring and managing performance against those goals.
- ▶ Core BPM processes include financial and operational planning, consolidation and reporting, modeling, analysis, and monitoring of key performance indicators (KPIs) linked to organizational strategy.

The business performance management addresses metrics, methods, processes, and associated systems used for the purpose of monitoring and managing the business performance of an enterprise. In particular, it:

- ▶ Encompasses balanced scorecards, operational dashboards, business activity monitoring, and budgeting/planning/forecasting

- ▶ Leverages business intelligence, systems dynamics, Sarbanes-Oxley (SOX), six sigma, value creation, process management, and organizational design
- ▶ Requires a holistic vision of how to align business decisions and activities with strategic and operational objectives, though implementation is typically along an incremental route-map

Summary

In this chapter we introduced business process management and the IBM business process management solution. We also touched on the relationship between business process management and business performance management.



Product overview

This chapter provides an overview of the core WebSphere process integration products Version 6.0.2. These products support the complete business process management life cycle with a modular approach.

WebSphere process integration products enable you to:

- ▶ Model, simulate, and analyze complex business scenarios quickly and effectively before they are implemented.
- ▶ Transform existing processes to be dynamic and adaptive to deliver cost effective business agility.
- ▶ Allow users to monitor the business processes they have implemented so that they can continuously make improvements to them.

In this chapter we introduce these products:

- ▶ WebSphere Business Modeler V6.0.2
- ▶ WebSphere Integration Developer V6.0.2
- ▶ WebSphere Process Server V6.0.2
- ▶ WebSphere Business Monitor V6.0.2

For each product we also provide information about the enhancements that came with Version 6.0.2.

WebSphere Business Modeler

IBM WebSphere Business Modeler (Figure 2-1) is a business process modeling tool that enables you to model, design, simulate, analyze, and generate reports for your business processes, integrate your new and revised process, and define your organizations, resources, and business items.

Accurate definition and modeling business processes is a critical factor in improving business performance. A business process is defined by interactions that occur between an organization's components and its environment as the organization pursues its business objectives. Business processes are often complex because of numerous incremental changes that are made in reaction to business circumstances. Without formal process documentation and process management systems, these process complexities can burden an organization with unnecessary hindrances and bottlenecks. A well-constructed business process model can help you locate and eliminate those hidden inefficiencies, costs and delays.

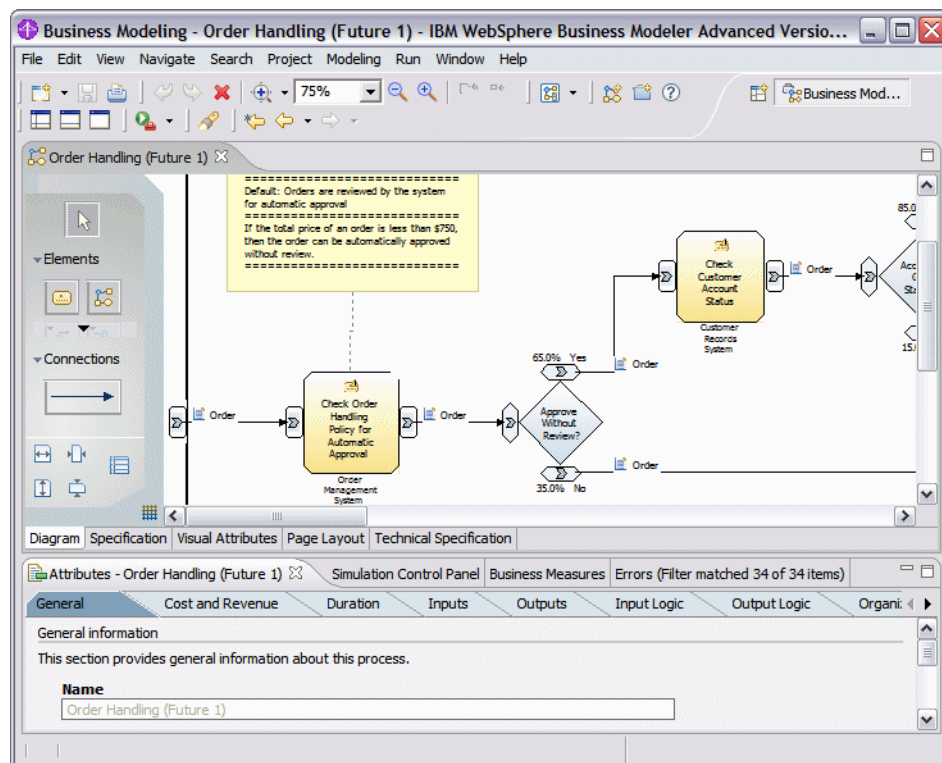


Figure 2-1 WebSphere Business Modeler

Complete and robust process modeling tool

A business process model is a visual representation of a process that contains supporting information. To create effective models, you must have a well-designed modeling structure that ensures consistent and complete representation of relevant information, including normal operations as well as alternatives and exceptions to standard procedures. You can use business process models to achieve many goals, including these:

- ▶ Documenting existing procedures
- ▶ Determining requirements for staff, systems, and facilities
- ▶ Planning changes to existing processes and systems
- ▶ Testing and analyzing existing and proposed processes

WebSphere Business Modeler enables you to:

- ▶ Transform business process for automation
- ▶ Separate your business process model from its underlying implementation
- ▶ Create business processes that are based on industry standards
- ▶ Leverage skill sets of business and IT professionals
- ▶ Collaborate with team members
- ▶ Simulate and analyze business process
- ▶ Improve communication between business and IT

Leverage skill sets of business and IT professionals

Whether you are an entry level or seasoned business analyst or an IT specialist, WebSphere Business Modeler provides the versatility to fit your particular set of skills. Using WebSphere Business Modeler, professionals with different scopes of interest and expertise can build process models to meet a wide range of business objectives. From the entry level business analyst, who requires an easy to use process mapping tool, to the seasoned business analyst, who requires sophisticated simulation capabilities to drive strategic decisions, to the integration developer who uses a process model as the framework for a new application, competitive businesses require a versatile modeling tool that has the flexibility to meet the needs of both business and technical professionals.

Collaborate with team members

WebSphere Business Modeler provides a built-in capability to connect with a separate product, 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 an enterprise and worldwide scale. 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.

Simulate and analyze business process

WebSphere Business Modeler provides a simulation function that lets you simulate and analyze your processes under any set of circumstances. When you simulate runtime processing, WebSphere Business Modeler provides an animated view of the business process in action. You can specify a wide variety of different conditions for the simulation, including the rate and composition of process inputs and the number of personnel and system resources available to handle the process. Through simulation and analysis you can quickly determine how the performance of your business process is affected in various real or hypothetical conditions.

Transform business process for implementation

WebSphere Business Modeler enables you to transform business process models to IT-level models. You can export a Business Process Execution Language (BPEL) version of a business process model, then use WebSphere Integration Developer to import the BPEL model and use this as a basis to create an application.

Improve communication between business and IT

WebSphere Business Modeler enables business analysts to define business processes in an unambiguous way. These processes can then be exported to WebSphere Integration Developer, which provides an excellent starting point for IT to implement the process.

Revolutionizes business flexibility

Because business environments are constantly changing, requiring continual fine-tuning of processes, business improvement is a perpetual race. WebSphere Business Modeler facilitates communication between business organizations by allowing you to create a process model that has far broader uses than a static drawing. Furthermore, WebSphere Business Modeler delivers cost saving benefits by providing a single tool that effectively utilizes the same process information for many purposes, reducing the duplication of effort required by using multiple, incompatible tools.

More information

For more information about WebSphere Business Modeler, refer to:

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

WebSphere Business Modeler Version 6.0.2

WebSphere Business Modeler Version 6.0.2 has many new functions and enhancements that increase the usability of the product.

New Business Measures view

The new Business Measures view replaces the Business Measures editor from Version 6.0.1. In the new Business Measures view, you can easily define all the business-level information needed for your monitoring requirements: The name and description of each metric and the name, description, targets, and ranges of each key performance indicator (KPI). You can also specify how the resulting information should be presented in the dashboard views of WebSphere Business Monitor.

The more complex and technical tasks that are typically performed by integration developers (such as specifying KPI calculations and expressions) have been removed from WebSphere Business Modeler and are now delivered as part of the new business measures tooling for WebSphere Business Monitor Version 6.0.2. The Monitor Development Toolkit can be used in WebSphere Integration Developer to create monitor models from scratch or to refine monitor models imported from WebSphere Business Modeler.

Process modeling enhancements

A number of improvements have been made to enhance the user experience when creating process models:

- ▶ Scroll in a process diagram when selecting multiple elements using the marquee selection tool.
- ▶ Reorder the input and output branches of decisions, forks, merges, and joins to rearrange the connections from or to the branches.
- ▶ Search for model elements in the Project Tree view by model element name at any level in the view (in V6.0.1, you could search top-level elements only).
- ▶ Search by description.
- ▶ Delete an unassigned swimlane from a process diagram if the swimlane does not contain any model elements.
- ▶ Repeat your selection from previous choices of many elements and attributes by pressing F3 or F4 in a window, for example, in the Type Selection or Select Role Requirement window.
- ▶ Set durations using simplified options that let you define the length of months and years as a number of days.

- ▶ View classifier attribute names and values on process diagrams.
- ▶ View the full path names in the data labels on process diagrams.
- ▶ In the swimlane layout, connections no longer overlap.

Simulation and dynamic analysis enrichment

Enhancements to WebSphere Business Modeler simulation and dynamic analysis capabilities include the following new functions:

- ▶ Ability to view, simulate, and animate a simulation run in swimlane layout.
- ▶ Improved model validation through the auto detection of missing stop nodes when you take a simulation snapshot.
- ▶ Options to control how task duration is affected by resource availability during simulation.
- ▶ Compacting of analysis information through the use of roll-ups for summary information.
- ▶ Option to limit the extent of process case matching for analyzing a process model that has a large number of possible cases.
- ▶ Ability to view the full path names in the data labels of the Simulation editor.

Reporting and printing improvements

New functions let you do more with the WebSphere Business Modeler reporting and printing:

- ▶ Define and apply a header or footer master to one or more report templates.
- ▶ Control the formatting of certain data types (date, duration, currency, integer, double, percentage, and Boolean) so that, for example, 2006-03-23T12:00:00-05:00 converts to March 23 2006 12:00 EST.
- ▶ Define global parameters (for example, a company name or logo) to multiple reports.
- ▶ Sort fields in the report designer by ascending or descending order to produce structured reports.
- ▶ Export profile and dynamic analysis reports in XML format for use with other tools.
- ▶ Include working and resource duration attributes in the Process Instance Summary and Process Instance Time dynamic analysis reports.

Import and export enhancements

You can now perform or specify the following imports and exports:

- ▶ Import business services (WSDL files) and business service objects (XSD files). Multiple files can be imported from different file directories or a service registry (WebSphere Service Registry and Repository). WebSphere Business Modeler creates representations of these files so they can be used when you are modeling processes. This information is also carried forward to WebSphere Integration Developer when exported.
- ▶ Set the implementation type for the technical attributes of tasks and services to import Web service, SCA, or JMS bindings and create import files with the corresponding binding.
- ▶ Import and export projects using the new MAR file extension.

Interoperability improvements

In addition to importing and exporting business process models, the following interoperability improvements in WebSphere Business Modeler streamline the transfer of information to and from other tools:

- ▶ Copying a diagram (or parts of a diagram) and pasting it into any program that enables the pasting of pictures (such as Microsoft® Word, Microsoft PowerPoint®, or Microsoft Visio®).
- ▶ Rational RequisitePro® integration for linking business requirements to your process models.

Publishing enhancements

You can now publish swimlane layouts of process diagrams and select which swimlane layouts you want to publish.

WebSphere Business Modeler Publishing Server

WebSphere Business Modeler Publishing Server Version 6.0.2 now has added support for RedHat Enterprise Version 4.0 and SuSE Enterprise Server 9.0 on Intel® 32 bit.

Other enhancements include:

- ▶ Ability to add an attachment at the same time that a comment is created.
- ▶ Ability to provide more feedback for unauthorized areas of Publishing Server.
- ▶ Support for URL hyperlinks.

WebSphere Integration Developer

IBM WebSphere Integration Developer (Figure 2-2) is the integration tool you use for all your process integration requirements. It allows you to build flexible, composite applications by wiring service components with minimal skills based on service-oriented architecture (SOA).

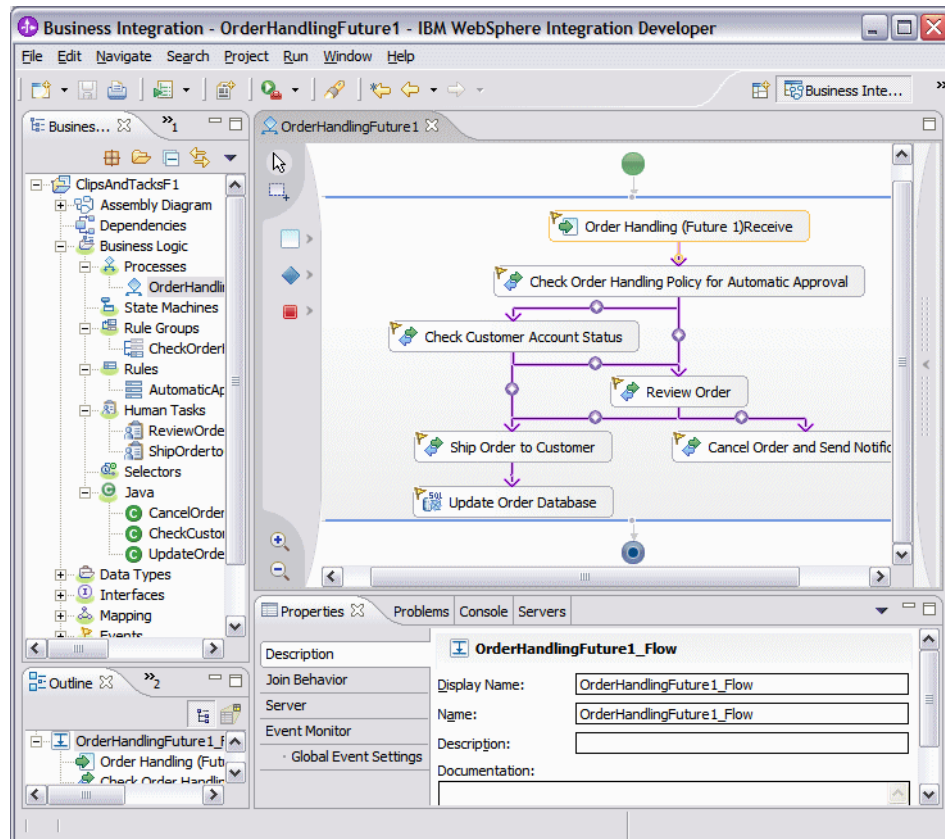


Figure 2-2 WebSphere Integration Developer

Eclipse-based development tool

IBM WebSphere Integration Developer, Version 6 software is Eclipse technology-based tooling designed to enable you to rapidly assemble business solutions based on a composite application development framework and using minimal programming skills.

With WebSphere Integration Developer, you can author SOA-based services and choreograph them into business processes that you can deploy on IBM WebSphere Process Server. WebSphere Integration Developer offers a role-based development experience that specifically targets the integration developer on a single and integrated Eclipse platform.

SOA and SCA

WebSphere Integration Developer is the tool for all your integration requirements and fully implements the SOA. SOA and the Service Component Architecture (SCA) enable you to convert and use your existing IT architecture as reusable service components. The framework is comprehensive and extensive, which helps you transform your enterprise to SOA by leveraging your existing IT architecture. Because WebSphere Integration Developer is based on standards-based technologies, such as Web Services Description Language (WSDL), XML Schema Definition (XSD) and Business Process Execution Language (BPEL), you can model, implement, and deploy complex composite applications without extensive knowledge of the underlying implementation.

WebSphere Integration Developer is a fully integrated development environment based on the Eclipse 3 platform. Its graphically rich interface allows developers to create composite business applications by wiring service components with minimal skills and agnostic of underlying programming implementations. It has a fully integrated testing, debug, and deployment environment that allows you to easily deploy to WebSphere Process Server. Once deployed, you can dynamically change and adapt to changes with its rich features, such as business rules, selectors, and state machines.

Integration with other development tools

WebSphere Integration Developer complements IBM WebSphere Business Modeler Version 6, and can be used in conjunction with IBM Rational Software Architect Version 6, IBM Rational RequisitePro, and IBM Rational Application Developer Version 6. When combined into a single integrated development environment (IDE), these products provide a complete suite of tools to model, simulate, author, and deploy composite SOA applications.

Business process support

WebSphere Integration Developer delivers a rich set of features to assemble, deploy, and manage business processes:

- ▶ Import business models and business items from WebSphere Business Modeler

- ▶ Drag and drop features to create business processes
- ▶ Widgets to wire service components, including:
 - Business processes
 - Human tasks
 - Business state machines
 - Business rules
 - Supporting services
 - Interface maps
 - Business object maps
 - Relationships
 - Selectors
 - Java objects
- ▶ Look up, import, and publish services from/to WebSphere Service Registry and Repository
- ▶ Import services from ERP and EIS systems, such as PeopleSoft, SAP, CICS, and IMS™

In addition, WebSphere Integration Developer enables:

- ▶ Composite application development:

WebSphere Integration Developer allows you to render existing applications as standards based services. Services can be assembled together without the associated complexities of the underlying IT required for complex business applications. Now, you start building a solution from the services available to you and build only the services that do not exist, enabling a true incremental solution.
- ▶ Flexibility for managing deployed processes:

WebSphere Integration Developer allows you to test and debug business processes, and deploy them on WebSphere Process Server with a few mouse clicks.

WebSphere Integration Developer's rich GUI intensive features allow for change and management of deployed processes with minimal skills and disruption.

More information

For more information about WebSphere Integration Developer, refer to:

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

WebSphere Integration Developer Version 6.0.2

WebSphere Integration Developer has been improved and new functionality has been added in several areas. The main points of improvement are outlined below. Please consult the product documentation for a complete list of enhancements.

Visual snippets

New categories of visual snippets have been added, converter and numbers, and more snippets have been added to the existing categories. Significant additions can be found in the SCA services category, for instance.

Rational RequisitePro Integration

Integration Developer now provides a RequisitePro view (Requirement view), which allows you to link a process component to a Requirement managed by RequisitePro. This enables a tighter integration between business and IT.

Top-down Web services development

When generating a Web service binding in the Assembly Editor, you can now generate a skeleton Web service project using the Web services wizard of Rational Application Developer.

Business rules

A number of enhancements have been introduced:

- ▶ An initialization rule can be specified for a decision table.
- ▶ A function to copy a business object has been added, which makes it easy to copy the input business object to the output, for instance.
- ▶ An Otherwise clause can be specified on conditions in a decision table.
- ▶ It is now possible to terminate the execution of a rule set early by using return. Previously a rule set would always evaluate all rule conditions.
- ▶ Auditing has been introduced, which allows tracking of all changes to business rules. A record of what was changed and who made the change can be logged.
- ▶ Import and export of rules is now possible, streamlining the process of moving rule definitions between different environments.

Business state machines

State machines can be queried for their current state, and support for management of state machines has been added in BPC Explorer.

Human tasks

Some of the improvements for human tasks include:

- ▶ Improved group support with dynamic group membership determination when a user logs on—previously this was done at the time of task creation.
- ▶ Support for ad-hoc collaboration through subtasks and follow-on tasks.
- ▶ Work items can be transferred between groups.
- ▶ Customizable e-mail notifications can be triggered on escalation. Notifications can contain variables from the task in the subject and body of the e-mail.
- ▶ New wizard to generate JavaServer™ Faces based Web clients for human tasks. This provides an excellent starting point for developing custom applications for human tasks.
- ▶ Server controlled page flow enables a smooth workflow where the same person has to claim and complete a number of consecutive steps within a process.
- ▶ A staff plug-in interface allows the result set to be modified after the query is executed. A usage pattern for this capability could be to manage the workload for users or enable substitute assignment if a person is away from work for a period of time.
- ▶ Support for binary custom properties for human tasks has been added. So now you can add scanned images or an audio recording of a conversation to a task.
- ▶ Work items can now be temporarily suspended.

Information service

The information service is installed separately with Rational Product Updater, and provides easy access to relational database systems with full support of SQL from within a business process.

Adapters

WebSphere Integration Developer now ships with all the JCA adapters:

- ▶ Three new adapters have been included: E-mail, FTP, and JD Edwards®.

- ▶ The technology adapters (flat file, FTP, JDBC™ and e-mail) are for development and production use.
- ▶ The application adapters (SAP, Siebel®, PeopleSoft, JD Edwards, and Oracle E-Business Suite) are for development use only.
- ▶ New discovery agent for WebSphere Service Registry and Repository, which enables discovery of existing services in the registry and publication of new services.

Monitor Development Toolkit

The WebSphere Business Monitor Development Toolkit provides the tools for creating monitor models that can be transformed into executable code for WebSphere Business Monitor. The toolkit runs inside WebSphere Integration Developer and includes the Monitor Model Editor and a unit test environment.

The Monitor Model Editor is a visual editor for creating monitor models. A monitor model describes business measures (such as metrics and key performance indicators), their dependencies on incoming events, conditions warranting business actions, and outbound events that report such conditions and might trigger business actions.

The WebSphere Business Monitor Development Toolkit also includes a unit test environment, which is a lightweight environment for testing the monitor model in WebSphere Integration Developer and requires no prerequisites.

The Monitor Development Toolkit comes with WebSphere Business Monitor and is installed into WebSphere Integration Developer.

Event definition editor

The event definition editor is a new editor installed as a part of the Monitor Development Toolkit, which allows you to create and edit custom common business events (CBE) for consumption by WebSphere Business Monitor.

Refactoring

Refactoring allows you to move and/or rename artifacts in Integration Developer. Dependencies are examined and you are given an option to update all references. The tasks that support refactoring are:

- ▶ Rename
- ▶ Change namespace
- ▶ Move to another library or module
- ▶ Extract embedded XSDs and WSDLs

WebSphere Process Server

IBM WebSphere Process Server (Figure 2-3) is a business integration server. It is built to support solutions based on service-oriented architecture (SOA). You can use it to build advanced business processes and traditional business integration such as enterprise application integration. WebSphere Process Server is based on WebSphere Application Server and contains the best of the product features previously found in WebSphere MQ Workflow, WebSphere InterChange Server and WebSphere Business Integration Server Foundation.

By building on top of WebSphere Application Server Network Deployment, WebSphere Process Server can take advantage of all the mature capabilities it provides, such as clustering, high availability, embedded messaging and transaction management.

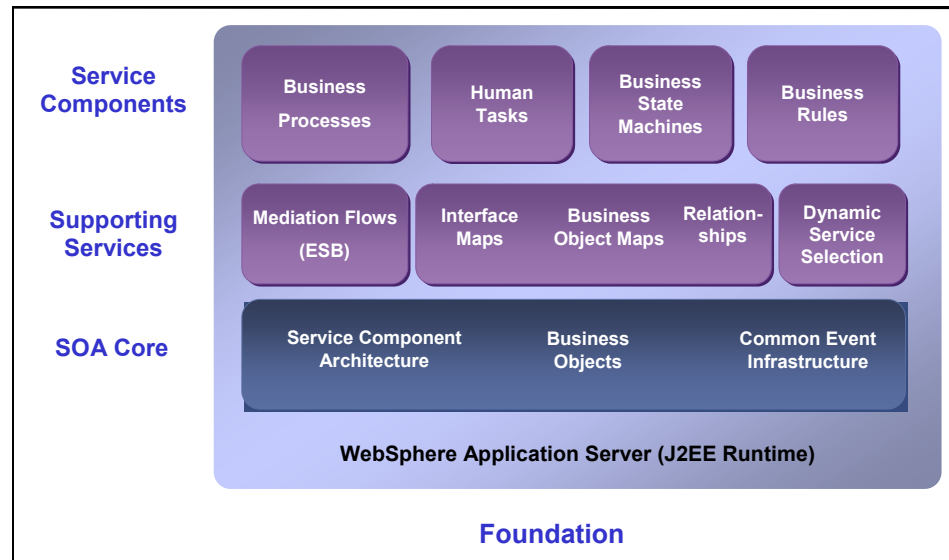


Figure 2-3 WebSphere Process Server architecture

WebSphere Process Server includes three layers:

- ▶ SOA core
- ▶ Supporting services
- ▶ Service components

SOA core

SOA core consists of Service Component Architecture (SCA), business objects, and the common event infrastructure (CEI).

Service Component Architecture (SCA)

SCA presents all elements of business transactions – access to Web services, Enterprise Information System (EIS) service assets, business rules, workflows, databases, and so on – as service components. SCA separates business logic from implementation, so that you can focus on assembling an integrated application without knowing the implementation details. Service components can be assembled graphically in WebSphere Integration Developer, and the implementation can be added later.

An SCA component needs to specify its interface, implementation, as well as references (Figure 2-4).

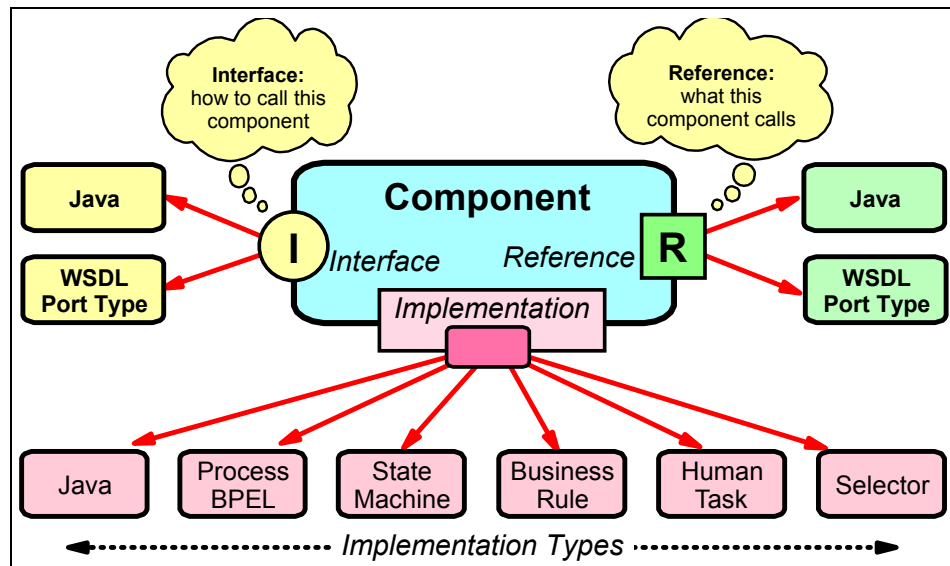


Figure 2-4 Service component

► SCA interface

By definition, an interface is the place at which independent and often unrelated systems meet and communicate with each other. An interface can be defined by any programming/interface definition language.

WebSphere Process Server currently supports a Java interface definition and an XML definition (WSDL port type). Arguments described in an XML Schema are exposed to programmers as SDO data objects.

The WebSphere Integration Developer tooling primarily generates interfaces using the WSDL representation.

► **SCA implementation**

The SCA implementation specifies the implementation type of the component's interface. Developers can implement business services in their language of choice (for example, Java, BPEL, or state machine).

Current implementation types include business process, human task, interface map, selector, business rules, business state machine, and Java.

► **SCA references**

An SCA reference specifies other SCA services that a component uses. These can be softlinks, which do not have to specify which specific component will be used.

Business objects

Business objects define the data flowing between different components in the Service Component Architecture. Business objects provide an abstraction for data access and is based on a data access technology called Service Data Objects (SDO). SDOs provide a universal means of describing disparate data.

Business objects provide rich features to map, manage and transform data to underlying IT and are described through standards based XML Schema (XSD).

Common event infrastructure (CEI)

The common event infrastructure allows service components to emit events that can be captured by business monitors such as WebSphere Business Monitor for real-time monitoring of business processes.

Common business events (CBE) is a proposed standard for how to capture basic business event information, such as the date/time an event is created, correlation IDs (relevant business events) in a unified way. IBM has submitted CBE to the Organization for the Advancement of Structured Information Standards (OASIS) for consideration as a new standard.

Supporting services

On top of the runtime infrastructure and the SOA core, WebSphere Process Server offers a variety of service components. Supporting services are components that are needed in any integration solution, including data transformation and synchronization services.

Mediation flows

Implementing the main features of WebSphere Enterprise Service Bus (ESB), a mediation component can be used to convert XML-based data formats. It can also be used to mediate between various transports, for example, JMS and Web

services. A mediation component contains a “message flow” that operates on a message including transformation, logging and filtering and custom Java operations. Above and beyond the ESB mediation component, WebSphere Process Server contains higher-level transformation capabilities.

Interface maps

Interface maps let you invoke components by translating these calls. It is possible for interfaces of existing components to match semantically but not syntactically (for example, *updateCustomer* versus *updateCustomerInDB2*). This is especially true for components that already exist and services that need to be accessed. Additionally, you can use business object maps to translate the actual business object parameters of a service invocation.

Business object maps

A business object map lets you translate one type of business object into another type of business object. You can use these maps in a variety of ways, for example, in an interface map to convert one type of parameter data into another.

Relationships

An object, such as a specific instance of a customer, will typically be identified in different systems using different keys, however all objects are representations of the same logical entity. As an example, you as a person will most likely have a social security number, an employee ID, and several customer numbers used with different companies, however, they are all different representations of the same logical entity, namely you.

In WebSphere Process Server relationships can be used to establish relationship instances between object representations of the same logical entity in disparate back-end systems. You may want to access the same logical entity within business integration scenarios, for example, the same customer’s address might need to be updated in various back-end systems, such as an ERP system and a CRM system. These relationships can be established and managed automatically using the Relationships service component.

These relationships are typically accessed from a business object map when translating one business object format into another.

Selector

A selector can be used for dynamic selection and invocation of different services, which all share the same interface. For example, an invocation of a service component to calculate a sales discount might need to vary depending on the season. During most of the year the standard discount calculation component is called, but during the holiday season and the following sales season alternative discount calculation components are required.

WebSphere Process Server offers a Web-based interface to enable dynamic updates to the selection criteria and target services, which means that a module that has been deployed at a later time can still be called by this selector component enabling dynamic changes to the integration solution.

Service components

WebSphere Process Server provides business processes, human tasks, business states machine, and business rules service components.

Business process

A business process component in WebSphere Process Server implements a Business Process Execution Language (BPEL) compliant process. You can develop and deploy business processes that support long and short running business processes and a compensation model within a scalable infrastructure. You can create BPEL models in WebSphere Integration Developer or import from a business model that you have created in WebSphere Business Modeler or any other modeling tool, which supports the BPEL standard.

Human tasks

Human tasks are stand-alone components in WebSphere Process Server that you can use to assign work to employees. Additionally, the human task manager supports the ad hoc creation and tracking of tasks. You can use existing LDAP directories (as well as operating system repositories and the WebSphere user registry) to access staff information. WebSphere Process Server also supports multi-level escalation for human tasks, including e-mail notification and priority aging. WebSphere Process Server includes an extensible Web client that you can use to work with tasks or processes. This Web client is based on a set of reusable JavaServer Faces (JSF) components that you can use to create custom clients or embed human task functionality into other Web applications.

Business state machines

Business state machines provide another way of modeling a business process. Some processes are easily described as a sequential flow of activities, and they can be modeled as business processes. However some processes are driven by events rather than a sequence, and in this case the business state machine is a better fit for modeling the process. One example would be an ordering process where you can modify or cancel the order at any time during the order process until the order is actually fulfilled.

Business rules

Business rules are a means of implementing and enforcing business policy through externalization of business function. This enables dynamic changes of a business process for a more responsive business environment.

Business rule authoring is supported with Eclipse-based desktop tooling. Business analysts can use the Web-based runtime tooling included in WebSphere Process Server to update business rules as business needs dictate without affecting other services.

Imports and exports

An import is a way of representing an external service in the SOA environment. For example, retrieving data from an ERP system can be turned into a service using the enterprise service discovery wizard. An export is the reverse, a way of representing an SOA service to an external service, such as a Web client.

Note: WebSphere Process Server also includes support for business-to-business (B2B) scenarios by including the IBM WebSphere Partner Gateway.

More information

For more information about WebSphere Process Server, refer to:

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

WebSphere Process Server version 6.0.2

In addition to runtime support for the enhancements described for WebSphere Integration Developer, WebSphere Process Server has been improved in the areas listed below.

Administration

New features in the administration area are:

- ▶ Improved SCA module administration. You can now modify Web service endpoints from the administration console making it easy to move modules between different environments.
- ▶ Mediation flow properties can now be viewed and modified via the administration console with immediate effect.
- ▶ Mediations can dynamically select endpoints at runtime, for instance via lookup in a registry such as the WebSphere Service Registry and Repository.

- ▶ Enhanced support for cleanup of completed processes.

Business Process Choreographer Explorer

The BPC Explorer has been enhanced as follows:

- ▶ BPC Explorer now shows a graphical view of processes, which enables you to track the status of a process instance.
- ▶ The filter criteria for searches have been improved and columns in the result are selectable.
- ▶ Custom views can be created based on the search definitions.
- ▶ Enhanced configuration and customization options.
- ▶ Enablement for state machines.

Business Process Choreographer Observer

BPC Observer is a Web application, which generates reports about the execution of business processes and human tasks. It shipped as a sample in version 6.0.1 but is now a part of the product. The BPC Observer enable you to:

- ▶ Observe state and evolution of processes, overall duration, and actual work time, for instance.
- ▶ Provide customizable reports and graphical charts of historical and accumulated data of processes.
- ▶ Retrieve statistical data on processes and activities via flexible drill downs.

Interoperability

The APIs for the Business Flow Manager and Human Task Manager are now exposed as Web services. Methods in the APIs are rendered as Web service operations. This means that clients capable of invoking Web services (such as .NET clients) are able to interact with the Business Flow Manager and Human Task Manager using a supported Web service API.

WebSphere Business Monitor

WebSphere Business Monitor is an application that is deployed and run on WebSphere Process Server. It displays dashboards, which consists of portlets, that enable you to monitor different aspects of business performance.

You can use the WebSphere Business Monitor to capture real-time, work-in-progress items and perform corrective actions by reassigning or suspending activities or processes. You can display real-time data from work items produced as the monitored process is running, and can also retrieve and view the historical data of the process.

Dashboards serve a wide audience, essentially all line-of-business and systems management users in addition to business executives, enabling them to perform these tasks:

- ▶ Monitor and manage business performance indicators.
- ▶ Retrieve information quickly and efficiently.
- ▶ Personalize the analysis and display of business performance reports, and compress information to focus on the business objectives and the key performance indicators (KPI).
- ▶ View business-critical information graphically, using visual cues such as color to improve the probability of timely problem determination and the speed of decision making.
- ▶ Visualize performance data such as KPIs and metrics, which may be summarized in reports and graphs.
- ▶ Analyze and investigate business situations by using drill-down capabilities to trace situations to individual events and inspect event details.
- ▶ Set up actions and alerts that are part of the management phase of a business performance management solution.

Each dashboard is composed of one or several data snapshots, referred to as views. The dashboards run within the WebSphere Portal Server environment, and for each dashboard, a portal page is created and a set of views (portlets) are laid out in the portal page. The dashboard administrator can set up the following types of dashboard views in WebSphere Business Monitor:

- ▶ **Active instances**—Display the values of all the business measures (KPIs, metrics, stopwatches, and counters) that you defined in the business measures model.
- ▶ **KPIs**—Display the values of KPIs relative to their acceptable limits (below limits, within limits, or above limits).
- ▶ **Gauges**—Display KPI values in the form of a gauge, like a speedometer or tachometer, relative to their acceptable limits or margins.
- ▶ **Alerts**—Display notifications that are sent when a business situation occurs.
- ▶ **Dimensions**—Provide a multidimensional view of business performance data. You can pivot on any defined business dimension to analyze different aspects of the historical performance.

- ▶ **Process diagrams**—Display a process model with visual cues showing the status of each run of the process.
- ▶ **Reports**—Display performance reports relative to a time axis. Such reports typically contain tables and graphs with textual descriptions summarizing the analysis.
- ▶ **Organizations**—Display the business organization units and their employees, in the form of a navigation tree. This information is stored on a user registry (such as LDAP) that WebSphere Portal is configured to use, and is not taken from WebSphere Business Modeler.
- ▶ **Export values**—Enables you to export the values resulting from the running processes to an XML file that can be imported by WebSphere Business Modeler.

More information

For more information about WebSphere Business Monitor, refer to:

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

WebSphere Business Monitor Version 6.0.2

With the Version 6.0.2 release, the monitor model life cycle has undergone a major revision. The Business Measures Editor in Business Modeler has been replaced by the Business Measures view, where business analysts can define KPIs on a non-technical level. The detailed specification of the business measures is performed using the Monitor Development Toolkit under Integration Developer.

Monitor Development Toolkit

The new Monitor Development Toolkit adds the capability for developers to define the technical aspects, such as triggers, counters, metrics, stopwatches, and measures of the new monitor model in WebSphere Integration Developer. A lightweight unit test environment (UTE) allows for easy test of the monitor model before deployment.

This also means that the requirement for the Modeler no longer exists as of Version 6.0.2. It is perfectly viable for business users to define KPIs in plain text to hand over to developers for implementation in the Monitor Development Toolkit. Additionally it is now possible to generate custom Common Base Events (CBE) from systems executing inside or outside the WebSphere Process Server environment, and these events can be consumed by WebSphere Business Monitor. This enables us to monitor basically any application that can produce CBEs.

Note: For a detailed discussion of the changes, refer to Chapter 3, “WebSphere Business Monitor Version 6.0.2 architecture” on page 39.

Summary

In this chapter we introduced the four products that are part of the IBM business process management solution. We will use the four products in our scenario to move a business process application from modeling to implementation to monitoring.

The online Information Center documentation of the four products can be found at:

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp>



WebSphere Business Monitor Version 6.0.2 architecture

In this chapter we describe the architecture of the WebSphere Business Monitor product, and define the associated terminology.

Note: If this chapter seems a bit too early in this book, you can skip over it for now and read it later when you get to Chapter 12, “Developing and testing the business measures with the Monitor Toolkit” on page 327.

Overview

WebSphere Business Monitor Version 6.0.2 is a Web-based client/server application that measures business performance, monitors activities and workflow, and reports on business operations. The information captured can help you identify problems, correct faults, and change processes to achieve a more efficient business.

WebSphere Business Monitor 6.0.2 monitors business activities at runtime by monitoring a set of events that contains information of interest to the business.

WebSphere Business Monitor calculates key performance indicators (KPIs) and metrics using collected events, based on a given model. The calculated KPIs and metrics values are represented on a number of views based on business needs. WebSphere Business Monitor notifies users of incidents requiring their attention and can also perform corrective actions to avoid failures. It supports different notification methods (alert, e-mail, cell phone, pager, and service invocation) in response to situations and actions associated with defined conditions.

WebSphere Business Monitor depends on business measures models for its monitoring procedure. These models are created in the Monitor Model Editor where you can specify the measuring points and event filters, define the measurements, their correlations, and sources of the business data. When the business measures model is complete, you can export it to WebSphere Business Monitor. It then recognizes the model to be monitored and the measurements to be captured from the incoming events.

You can use the Monitor Model Editor to open the process models created in WebSphere Business Modeler and to create monitoring models. For each monitoring model, you can define the metrics and KPIs, event emission points, event filters, event composition rules, and situations that will trigger specific actions at runtime.

Here are some key features of WebSphere Business Monitor 6.0.2:

- ▶ Captures a large amount of data through events from operation activities and transforms it into metric and KPI values
- ▶ Extracts the measurements variables from business data
- ▶ Displays the measurements values in useful views
- ▶ Provides analysis and reports
- ▶ Notifies users to take action to prevent failures
- ▶ Supports monitoring any kind of business activities as long as it can submit events to the Common Event Infrastructure (CEI)

This chapter includes the following sections:

- ▶ Overview of the Modeler to Monitor closed loop cycle
- ▶ WebSphere Business Monitor architecture overview
- ▶ Overview of the Monitor Dashboard

Note: Software installation, prerequisites, and configuration of Monitor V6.0.2 required prior to model deployment is discussed in detail in Appendix B, “Installation of WebSphere Business Monitor” on page 619.

Overview of the Modeler to Monitor closed loop cycle

In this section we describe the process of taking the business process model from WebSphere Business Modeler V6 to WebSphere Business Monitor V6.

Figure 3-1 shows the overall closed loop process from Modeler V6.0.2 to Monitor V6.0.2, as well as the supporting tooling and runtime engines for the business process:

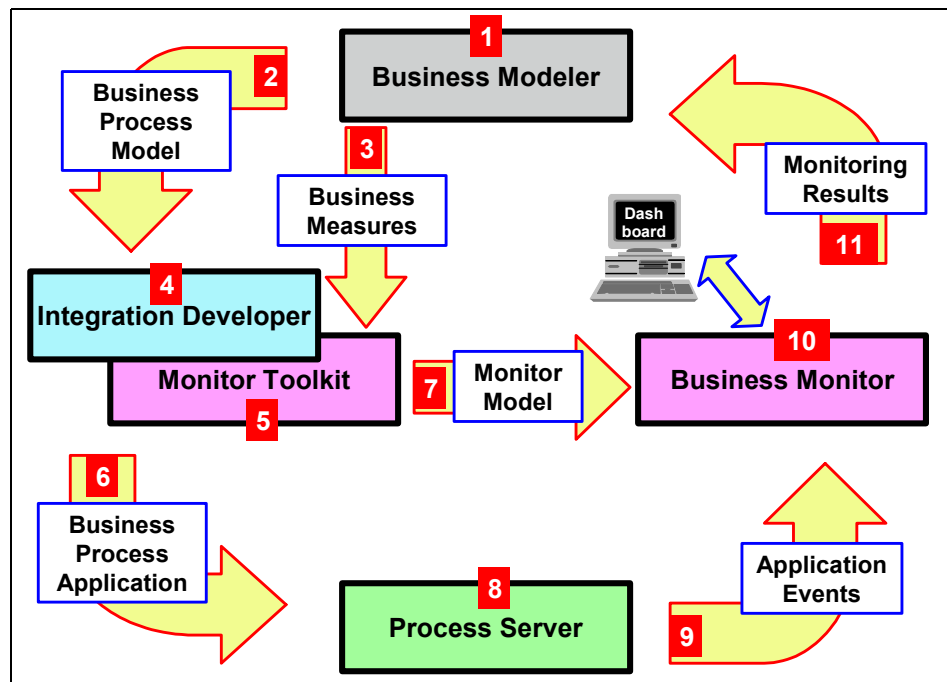


Figure 3-1 Closed loop process: Modeler V6 to Monitor V6

1. The business process model is developed in the Business Modeler.
2. The business process model is passed to Integration Developer in Business Process Execution Language (BPEL) format.
3. The business measures to be monitored and the feedback measures from the Monitor are identified by the business analyst and passed to the Monitor Development Toolkit as an initial monitor model.
4. The business process is completed as an enterprise application in the Integration Developer using the service component architecture (SCA) and service data objects (SDO).
5. The base monitor model and the event definitions are generated from the process. The monitor model is completed in the Monitor Development Toolkit using the business measures identified in the Modeler.
6. The business process application is exported and deployed to the Process Server.
7. The monitor model is exported and deployed to the Business Monitor.
8. The business process is executed in the Process Server and events are generated.
9. The events are passed from the Process Server to the Business Monitor.
10. The Business Monitor processes the events and applies the business intelligence rules provided in the monitor model. The Monitor Dashboard is used by a business analyst to analyze the data.
11. After the process has been executed and the data has been processed, the data identified in Step 3 above is fed back to the Business Modeler to enhance future simulation and analysis.

The last step is referred to as the *continuous process improvement cycle*. This methodology allows businesses to reconcile their *as-is* business process with the *to-be* business process with actual collected runtime data.

This data can be compared to the simulations that were performed in the Modeler and hence, allows us to optimize the as-is business process model toward a more efficient to-be model.

In Part 5, “Continuous process improvement” on page 487, we explore the continuous process improvement life cycle by utilizing IBM WebSphere Business Modeler and Monitor with our sample process scenario. We demonstrate how this methodology works and how it will yield higher efficiency in the overall practice of business performance management (BPM).

WebSphere Business Monitor scenarios

In our sample scenario, WebSphere Business Monitor is used to monitor and analyze a business process, which has initially been modeled in WebSphere Business Modeler, implemented in WebSphere Integration Developer, and runs on WebSphere Process Server. Through modeling, implementing, and deploying the business process, then subsequently monitoring it, we demonstrate how the Monitor integrates well with the IBM WebSphere Business Process Management software portfolio.

WebSphere Business Monitor is built on the concept of event monitoring. Event monitoring implies that if a system generates events representing changes in status, then these could be monitored using WebSphere Business Monitor. The terminology used to describe this is called **Business Activity Monitoring (BAM)**.

Consider the situation where we have a database that represents the status of an activity that is performed in an organization. When an order is received, a record is created in the database, and when it is shipped, the same record is updated. We can create a component that emits events representing the status of the record in the database. This can then be modeled in WebSphere Monitor, allowing us to collect the generated events and perform any business intelligence analysis required.

Therefore, we begin with a brief discussion of the different scenarios in which WebSphere Business Monitor can be used, in combination with the other tools in the Business Process Management suite, where appropriate (Figure 3-2).

	Modeler	WID / WPS	Monitor
1. BAM for BPEL (w/ Modeler)	✓	✓	✓
2. BAM for BPEL (w/o Modeler)		✓	✓
3. BAM for the rest of us			✓
4. General BPM	✓		✓

Figure 3-2 Scenarios for using WebSphere Business Monitor

1. **BAM for BPEL (with Modeler)**—Use this scenario if you have a BPEL process that has been created in the Modeler, then implemented in Integration Developer, prior to deployment on WebSphere Process Server.

This is the case for the sample business process used in this book. Use this option if you want to take advantage of the round trip capabilities discussed in “Overview of the Modeler to Monitor closed loop cycle” on page 41.
2. **BAM for BPEL (without Modeler)**—Use this scenario if you want to directly implement your BPEL process in WebSphere Integration Developer and carry out all work on the business measures model within the Monitor Development Toolkit plug-in.
3. **BAM for the rest of us**—Use this scenario if you want to monitor a non-BPEL process (not running in Process Server), which has been developed independently of WebSphere Business Modeler. Events from non-BPEL applications are suitable for consumption by Monitor if they conform to the CBE 1.0.1 specification.
4. **General BPM**—Use this scenario if you want to monitor a non-BPEL process (as in option 3, not running in Process Server), which has been modeled using WebSphere Business Modeler. As with option 1, this scenario also allows you to take advantage of the round trip capability between Modeler and Monitor for continuous process improvement.

WebSphere Business Monitor architecture

Figure 3-3 summarizes the main internal and external components in a WebSphere Business Monitor environment.

Let us first understand how the Monitor is working:

1. The **Monitor Development Toolkit** is shipped with WebSphere Business Monitor and is installed as a plug-in in WebSphere Integration Developer. Using the toolkit, a monitor model is developed based on events of a business process or other application. The monitor model is deployed on a Monitor Server.

An initial model can optionally be created using the Modeler.
2. The business process or other application runs in a server (**Process Server** or other) and generates events, based on the common base event (CBE) definition.
3. The **Common Event Infrastructure** (CEI) receives the events and sends them to the Monitor Server.
4. The monitor model executes in the **Monitor Server**. Based on the events, instance data is stored in the MONITOR database.

5. Certain conditions trigger situation events that are sent back to the CEI.
6. The CEI sends the situation events to the **Action Manager**.
7. The Action Manager reacts to these events and triggers alerts, phone calls, or e-mails.
8. The MONITOR database state information is replicated to the DATAMART database.
9. The **Monitor Dashboard** provides clients with aggregated information in a number of portal pages with Monitor portlets.
10. The Monitor Dashboard can export actual values (activity duration and decision percentages) to the Modeler.

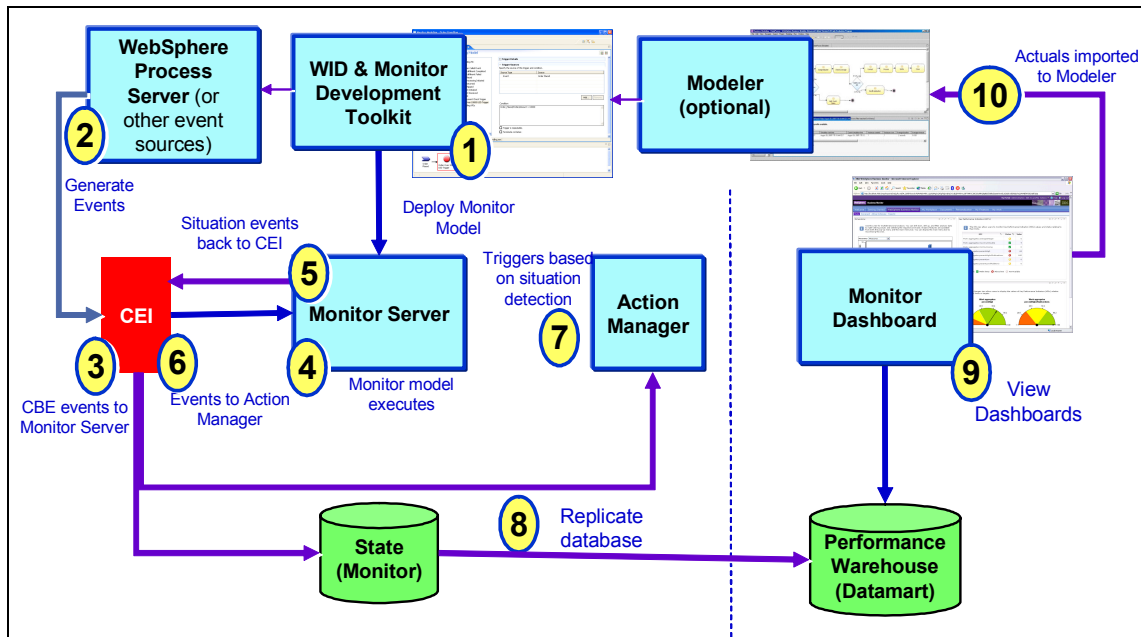


Figure 3-3 WebSphere Business Monitor V6.0.2 architecture

Monitor components

WebSphere Business Monitor consists of two servers and two databases. The Monitor Server is built on top of Process Server and the Dashboard Server is built on top of WebSphere Portal Server. The two databases are the MONITOR database for active instances and the DATAMART database for analysis of historical data.

In this section we look in more detail at the architecture of WebSphere Business Monitor, beginning with a discussion of the Monitor Development Toolkit.

Monitor Development Toolkit

The Monitor Development Toolkit is integrated as part of WebSphere Integration Developer. It introduces a new *Business Monitoring* perspective within the Integration Developer. The Monitor Development Toolkit has two components, the Integrated Development Environment (also referred to as the Monitor Model Editor) and the unit test environment (UTE).

Monitor Model Editor

The Monitor Model Editor assists the user in developing monitor models and creating event definitions. The monitor model can be based on a model imported from the WebSphere Business Modeler, or can be created from scratch using the Monitor Model Editor. The Monitor Model Editor exposes the Monitoring Programming Model with a graphical user interface to aid the user in their development effort (Figure 3-4).

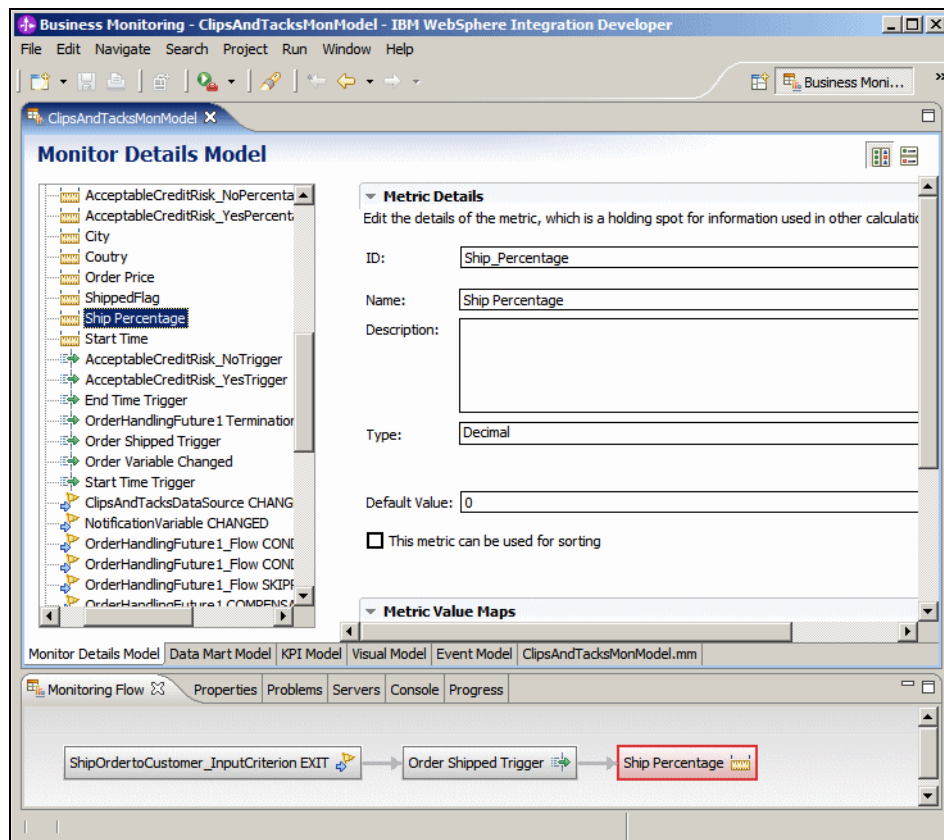


Figure 3-4 Monitor Development Toolkit: Monitor Model Editor

Monitor model terminology

In this section we define a number of terms associated with the monitor model. This terminology is used when referring to the monitoring model built in the Monitor Development Toolkit, and also when discussing the information stored in the two databases of the Monitor.

Figure 3-5 illustrates possible flows between these items.

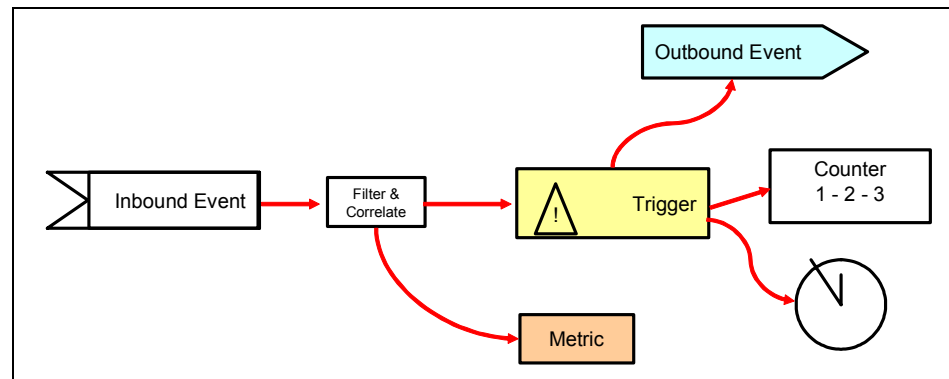


Figure 3-5 Monitor model components

For example, an inbound event emitted by the business process passes through a filter. Because we have specified an interest in some events, this inbound event causes a metric to be updated. The event also fires a trigger. There are a number of items that depend on this trigger, for example, an outbound event can be generated, which in turn results in an alert being sent to the business analyst. Alternatively, the trigger can start a stopwatch or increment a counter.

Monitor details model

The monitor model is composed of a set of different models; the first of these is the monitor details model. The monitor details model provides a high-level representation of the different components that are intended to represent the data to be analyzed at the process instance level. This model contains the information required by the other models to perform further analysis. The components of the monitor details model are shown in Table 3-1.

Table 3-1 Monitor details model

Details model	Description
Monitoring context	A monitoring context is a construct used to group a set of metrics, counters, stopwatches, triggers and event definitions for a specific instance. The monitoring context definition specified in the Monitor Details Model tab of the Monitor Model Editor defines the structure of the monitoring context instances that are created at runtime. Specific monitoring context instances are typically created through inbound events and terminated through a trigger that has the <code>terminateContext</code> attribute. Monitoring context definitions can be nested to provide enhanced visibility and correlation between different metrics.
Metric	Metrics represent the information that is required to be monitored through the life cycle of a monitoring context. Metrics are updated as a result of triggers and can also be updated with data contained in inbound events.
Key	A key is a specific type of metric. It is a unique identifier for the monitoring context instance and is used to correlate runtime events with specific instances. This ensures that events are delivered to the correct monitoring context instance. An inbound event will create a new monitoring context if there is no monitoring context instance associated with this event.
Counter	A counter is a specialized metric that can be incremented, decremented, or reset to zero. Counters can be used to track occurrences of specific events and are updated as a result of triggers or inbound events.
Stopwatch	Stopwatches are another type of specialized metric and can be started, stopped, or reset by a trigger or an inbound event.
Trigger	Triggers are used to detect a specific situation and then cause specific processing to occur in response to this situation. There are two aspects to a trigger. First, the <i>trigger source</i> specifies when to evaluate the trigger. The trigger source could be the change in value of a metric, arrival of an inbound event, another trigger firing, or it could be a periodic trigger that evaluates, for example, once per hour. Second, the trigger definition can optionally specify a <i>trigger condition</i> which determines whether the trigger fires when it is evaluated. This is a boolean condition that depends on specific attributes in either the monitoring context or data carried by the inbound event.

Details model	Description
Inbound event	Inbound events are the events that the business process emits and that are processed by the Monitor to generate the required monitoring information. An inbound event definition defines an event entry point in a monitoring context for receiving events of this type. By specifying a filter, it defines the events that the monitoring context receives. The event type and filter expression together define the event subscription. Inbound events can cause all the same effects as a trigger, except for terminating the monitoring context.
Outbound event	An outbound event definition specifies an event that can be generated by the monitoring model and the values it should contain. These situation events generated by the Monitor are processed by the Adaptive Action Manager and used to generate alerts. Outbound events can be emitted as a result of an incoming event, a situation detection, or on a periodic basis. The outbound event can be specified to contain business data in its extended data elements. Outbound event definitions define an event emission point in a monitoring context and the conditions for event emission.

Data mart model

The different types of analysis that the Monitor can perform provide a business intelligence perspective of the historical data collected across all instances.

The data mart model defines the aggregation analysis that can be performed across the instances that were modeled in monitor details model (Table 3-2).

Table 3-2 Data mart model

Data mart model	Description
Cube	A cube provides a dimensional structure to instance data. Cubes contain facts, measures, and dimensions. To set up a cube, you specify the facts that you want to track, the aggregation function to apply to each fact to create a measure (such as average, sum, count), and the dimensions you are interested in. The cube then provides the facility to filter instance data by specific metrics (dimensions), while also aggregating other metric values across instances (measures).
Fact	Facts contain metrics for the monitoring context. By default, there is a fact for each metric which is not specified as a dimension. Typically facts have quantitative data types so that they are suitable for aggregation across instances to construct a measure.

Data mart model	Description
Measure	A measure references a fact and adds an aggregation type. While a metric has a separate value per instance, measures aggregate these values across all instances, according to the aggregation function specified. The aggregation functions available are sum, minimum, maximum, count, and average.
Dimension	Dimensions enhance the capabilities of analyzing historical data by organizing it based on the values of specific metrics. Dimensions allow us to filter the measures based on the values of selected metrics. Multiple groupings can be applied simultaneously. For example, we could display orders submitted in March (time dimension) by customers in the USA (location dimension). Dimensions can have multiple levels, thus providing a hierarchy for drilling up and down through the data. For example, the location dimension may have a top level attribute of country, with city beneath it, enabling us to see orders in March, from New York.

Cubes and the associated constructs provide a powerful mechanism for analyzing historical data to identify trends and highlight problem areas. Multidimensional queries can be executed and we have the facility to drill up and drill down on the information, as required.

KPI model

The key performance indicators (KPI) model (Table 3-3) represents some specific measures that are selected as high-level KPIs. It uses the details and data mart models to provide the data, then adds ranges and targets for the KPIs. Further analyses can be specified and situation events can be defined.

Table 3-3 KPI model

KPI model	Description
KPI context	The KPI context purely serves as a container for KPI definitions, and trigger and event definitions that are defined at the KPI level.

KPI model	Description
KPI	Key performance indicators (KPIs) are quantifiable measurements of the improvement or deterioration in the performance of an activity critical to the success of a business. KPIs are based on a specific cube. Prior to building the monitor model, the appropriate KPIs for your business should be determined. The structure of your Monitor Details Model and Data Mart Model are determined by the KPIs you want to construct. KPIs provide you with the facility to aggregate measures over a specific set of instances. This can be done by filtering the KPI by a dimension. This dimension must be in the same cube as the business measure being considered. KPI targets and ranges can be specified in the monitor model, but also updated at runtime via the Dashboard. KPI values can be used to trigger business situation events, which can in turn generate alerts.

Unit test environment

A unit test environment (UTE) is also provided as part of the Monitor Development Toolkit (Figure 3-6).

The UTE provides the facility to test the monitor model prior to deploying it on the Monitor Server and configuring the Dashboard. The UTE is a lightweight environment that allows us to test key components of the model, without carrying out a full deployment. It also removes the pre-requisites (Portal, DB2® Alphablox, and DB2 Cube Views™) that are necessary for using the Dashboard.

Thus the UTE allows you to quickly deploy the monitor model and business process on a Monitor Server within the WebSphere Integration Developer environment. Metric values are displayed so that you can verify your model. The UTE is limited to displaying metrics, but this is sufficient to allow us to verify that the values of the metrics are collected as anticipated. The Dashboard is required to view aggregated data, such as dimensional measures and KPIs, but by testing the model in the UTE, metric errors can be identified early and rectified prior to deployment.

The screenshot shows the WebSphere Business Monitor interface. At the top, there is a header with 'WebSphere Business Monitor' and the IBM logo. Below the header, there are tabs for 'Active Instances', 'Configure', and 'View'. The main content area is titled 'Active Instances Values' and contains a table with the following data:

Order City	Order Country	Shipped Flag	Order_Count	Order Price	Shipped Order %	Approve With Out Review %	Start Time	Order Process Time
Markham	Canada	SHIPPED	1	214	100	100	Mar 5, 2007 3:10:10 PM	0d, 0h, 2m, 5s, 921ms
Markham	Canada	NOT_SHIPPED	1	1,569	0	0	Mar 5, 2007 3:11:32 PM	0d, 0h, 1m, 16s, 985ms
Markham	Canada	SHIPPED	1	2,353	100	0	Mar 5, 2007 3:11:48 PM	0d, 0h, 1m, 7s, 344ms

At the bottom of the table, there is a 'Refresh' button.

Figure 3-6 Monitor Development Toolkit: Unit Test environment

WebSphere Business Monitor components

In this section we discuss the major components that represent WebSphere Monitor. The components are split in such a way that decouples the event processing from the display of business measures in the Dashboard. The various components communicate with each other through common databases and components of the Enterprise Service Bus (ESB), namely the service integration bus (SI bus) and Common Event Infrastructure (CEI).

Monitor Server

Once the model has been tested in the Monitor Toolkit test environment, it can be exported as an EAR file for deployment on the Monitor Server. The Monitor Server is hosted on WebSphere Process Server 6.0.2 and consists of two components, the Monitor Model Manager and the Adaptive Action Manager.

Monitor Model Manager

The Monitor Model Manager acts as the execution engine for the monitor model. The *monitoring enabled* application generates a series of events. Based on the information provided in the monitor model, the Monitor Model Manager receives and extracts the required information from those events and updates the metrics as specified in the monitor model. The Monitor Server then updates the data stored in the MONITOR database accordingly. Following replication between the MONITOR and DATAMART databases, the relevant data is copied into the DATAMART database ready to be displayed on the Dashboard.

The Monitor Model Manager handles all of the metric calculations and business situation detections that are required for business process management. The Monitor Model Manager consumes events from the CEI and uses information from these events to update the metrics and KPIs displayed in the Dashboard views. It also sends outbound events to CEI, as defined in the monitor model.

Adaptive Action Manager

The Adaptive Action Manager is a service in the Monitor Server that processes the situation events emitted by monitor models running on the Monitor Server.

The Adaptive Action Manager is configured using the Monitor Server administrative console where the user creates bindings between situation events and the desired response.

The Adaptive Action Manager automates what should be done in response to outbound events reporting on the occurrence of a particular business situation. The CEI sends these events to the Adaptive Action Manager, which parses them, selects appropriate actions based on predefined rules created by the user, and invokes the selected action or set of actions. Consequently, when a situation event is received, the Adaptive Action Manager can invoke one or more action services in response.

The Adaptive Action Manager can perform two types of actions:

- ▶ Notification actions—These include sending an e-mail, SMS, pager message, or a dashboard alert. The dashboard alerts can be transmitted to all users, or a subset of users.
- ▶ Service invocation actions—These include invoking a Web service, or invoking a BPEL process through a Web service invocation.

Note: The Adaptive Action Manager uses LDAP as the user registry when sending notifications.

Common Business Event (CBE) source

Any application that emits events conforming to the CBE 1.0.1 specification can be monitored. For example, a BPEL process may be implemented in WebSphere Integration Developer and configured to emit events for monitoring. When it is deployed and starts executing, different events will be emitted, allowing the monitor to perform the required analysis.

Common Event Infrastructure (CEI)

The CEI is a component of the WebSphere Application Server or WebSphere Process Server. It provides the infrastructure for handling events and distributing them to different applications based on filtering criteria.

The system being monitored sends events to the CEI. These events are then consumed by the Monitor. Part of the monitor model definition involves specifying inbound events, including event filters that define the inbound events that the monitor model will consume. Similarly, the Monitor Server also emits outbound events to the CEI, which are consumed and processed by the Monitor Server Adaptive Action Manager.

The CEI provides facilities for the run-time environment to persistently store and retrieve events from many different programming environments. Events are represented using the common base event model, a standard, XML-based format that defines the structure of an event. The events are passed through JMS across the service integration bus (SI Bus).

Monitor databases

WebSphere Business Monitor V6.0.2 uses two databases, the MONITOR database and the DATAMART database. The DB2 replication manager copies data between these two databases. The replication manager uses the DB2 transaction logs to ensure that the data is replicated without contention.

MONITOR database

The MONITOR database consists of two sets of tables, the Repository tables and the State tables:

- ▶ The Repository tables contain metadata describing the deployed models. The tables also contain details regarding the location of the DATAMART database. The repository metadata is not replicated by the DB2 replication scripts.
- ▶ The State tables contain information on all running process instances, including the metrics for each instance. The Monitor Server stores event entries regarding the running process instances in the State tables. An event entry is the event data that is received so that a specific monitoring context can be updated.

Data from the State tables is transformed and moved to the DATAMART database by DB2 replication scripts. The only transformation performed is associated with the different performance requirements for the two databases. The MONITOR database is optimized for insertion, whereas the DATAMART database is optimized for reporting.

DATAMART database

The DATAMART database contains information on all completed and running process instances. It is populated by the replication scripts with data from the MONITOR database State tables. The Action Manager also stores data in the DATAMART database, which is then used in the Alerts view of the Dashboard.

The Dashboard queries the data in the DATAMART database using DB2 Alphablox. The Dashboard extracts metadata regarding the monitor model dimensions, metrics, and cube names from the repository component of the MONITOR database so that it can construct these queries.

The DATAMART database stores its data as star schemas (also referred to as cube tables), in contrast to the MONITOR database, where data is stored in flat form. The DATAMART database is optimized for aggregated and long running querying used by DB2 Alphablox in Dashboard views to provide advanced multidimensional reports.

Structure of the Repository tables

Table 3-4 highlights some of the Repository tables that are created in the MONITOR database when you deploy a monitor model.

Table 3-4 Summary of data in MONITOR database Repository tables

Repository table name	Purpose
META_MONITOR_MODEL	Displays models by version
META_CUBE	Displays a cube for each monitoring context
META_MONITOR_CONTEXT	Lists the monitoring contexts
META_CONTEXT_RELATIONSHIP	Displays the parent/child relationships between monitoring contexts
META_MONITOR_METRIC	Displays the metrics by monitoring context
AGGREGATED_MEASURE	Details the values flagged for export to Modeler
DBMAPPING	Details the mapping between artifacts and their database row/column names; this table is used to check for name collisions

Repository table name	Purpose
META_KPI	Contains KPI definitions
META_KPI_DIMENSION	Contains dimension information for the KPIs
META_KPI_RANGE	Details the ranges specified for the KPI
META_KPI_CONTEXT	Container for KPIs

The metadata in the Repository tables includes KPI targets and ranges that are queried by the Monitor Server to determine when to send business notifications. These values can be updated at runtime through the Dashboard.

Structure of the State tables

A State table is created in the MONITOR database with prefix CTX_ for each monitoring context. These tables are accessed and updated by the Monitor Server EJBs. They contain one column for each metric in the monitoring context and two to three columns for each stopwatch.

Additional columns are used to flag whether we have a repeating situation and to store the creation and termination times for the monitoring context instance. Each row in these tables represents one particular instance of the monitoring context.

Structure of the DATAMART tables

The DATAMART database contains a fact table with prefix FCT_ per monitoring context. These tables have one column per fact metric. The columns in the tables represent the facts contained within the cubes, as defined in the data mart model using the Monitor Development Toolkit.

The DATAMART database also contains a dimension table with prefix DIM_ per non-time dimension. There is then a DIM_TIME table which is shared by all cubes. A cube contains facts, measures, and dimensions. These cubes provide the Monitor Dashboard with the ability to filter instance data by specific metrics, and aggregating metric values across instances.

Within the DB2 Control Center, opening the Views section of the DATAMART database gives you a runtime view of the data, and you can see views with prefix CTR_ for each cube. This is a database view, not a real table, but illustrates the data used to construct the instances view in the Monitor Dashboard.

Monitor Dashboard

WebSphere Business Monitor provides a set of dashboards that can be used to display the processed data. The list of dashboards is as follows:

- ▶ Instances view
- ▶ Key performance indicator view
- ▶ Organizational view
- ▶ Report view
- ▶ Alerts view
- ▶ Diagram view
- ▶ Dimensional view
- ▶ Export values view
- ▶ Gauge view

Instances view

The Instances view displays the instances that were created for a specific monitoring context, providing the ability to display all the metrics that were created for that monitoring context (Figure 3-7).

This view could be wired to other portlets, allowing information on a specific instance to be passed to that portlet. For example, if it is required to stop a specific process instance, then a portlet could be developed that interfaces with business process container. This could be configured such that, when the process is selected in the instances view, data is passed to the new portlet containing all the information required to identify the process and stop it.



The screenshot shows a web interface titled "RoadRace MC" with a table of race results. The table has columns for Administration, Diagram, RunnerID, RaceDate, Mile01Duration, and Mile02Duration. The data is as follows:

Administration	Diagram	RunnerID	RaceDate	Mile01Duration	Mile02Duration
		aagomez	Jul 26, 2006 12:00:00 AM	5 m	11 m, 40 s
		aloboa	Jul 26, 2006 12:00:00 AM	10 m	23 m, 20 s
		dopyjoe	Jul 25, 2006 12:00:00 AM	13 m, 20 s	26 m, 40 s
		ferrell	Jul 27, 2006 12:00:00 AM	10 m	20 m
		millers	Jul 24, 2006 12:00:00 AM	20 m	36 m, 40 s

Figure 3-7 Instances view

Key performance indicator (KPI) view

The KPI view displays different aggregations on the data, thus providing different options for representation of data.

The KPI view displays the measures (average, sum) captured across different instances of the process or activity and displays them relative to a target and set of ranges. The target and ranges are editable at runtime.

Icons can be used to represent a specific range for an easier visualization of the data. For example, if the number of cancelled orders is greater than 100, then it should be displayed in a red color to highlight the fact to the viewer (Figure 3-8).

KPI Name	↑↓	Status	Value	Target	Value in Range
MaxRaceDurationInRaleigh			0 D, 0 H, 46 M, 40 S	0 D, 0 H, 45 M, 0 S	
Average Claim Amount in New York			674.17	350	
Total Claim Amount in New York			4045	4045	
My KPI2			42.85	24	
My KPI3			93.62	45	
kpi1			27.20	25	

Figure 3-8 Key performance indicator view

Gauge view

The Gauge view displays the same data as the KPI view, but show it in a gauge representation (Figure 3-9). This gives a better visualization for spotting a problem in the values of the key performance indicator values.

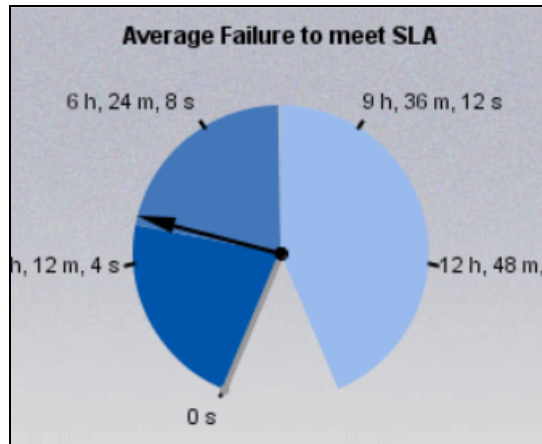


Figure 3-9 Gauges view

Organization view

The Organization view displays the structure of the organization (for example, organization units and employees) as defined in the user registry. This requires a user registry to be available, and any user registry that supports JNDI interfacing is supported.

This view provides a set of searching facilities that provide the user with capabilities for locating an employee and identifying the organization.

Reports view

The Reports view displays historical changes in dimensional measure values over a time period. It is for time-based analysis (Figure 3-10).

For example, if the monitoring model created contains a measure that represents the total number of orders received, the Reports view can be used to display the total number of orders received on daily, monthly, or annual basis.

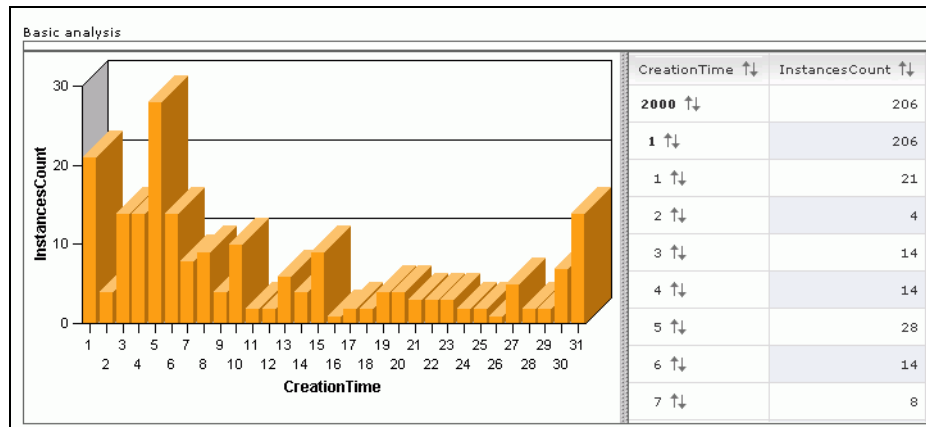


Figure 3-10 Report view

Alerts view

The Alerts view displays the alerts produced by the Action Manager (Figure 3-11). When the Action Manager receives the events indicating that a situation occurred, the associated alerts are delivered to the dashboard alerts view, assuming that the Action Manager was configured with an alert destination of dashboard.

<input type="checkbox"/>	Date and Time	Subject	Alert Source
<input type="checkbox"/>	Mar 7, 2007 3:16:45 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 7, 2007 3:47:34 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 7, 2007 4:00:59 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 7, 2007 4:07:07 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 8, 2007 12:17:37 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 8, 2007 12:17:37 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 8, 2007 12:17:37 PM	Order exceeded target shipping time	

Figure 3-11 Alerts view

Diagram view

The Diagram view can be used to display instance diagrams or other diagrams representing the monitoring context or KPI context (Figure 3-12).

The Diagram view can be used to view aggregated data across the entire set of process instances that are created. For aggregated data to be displayed in the Diagram view, it is necessary to first create a KPI that represents the data required, and then associate it with the Diagram view.

The Diagram view can also be used to view data based on the information for an individual process instance. This usage is for displaying metrics at the process instance level and associating the metric with the diagram.

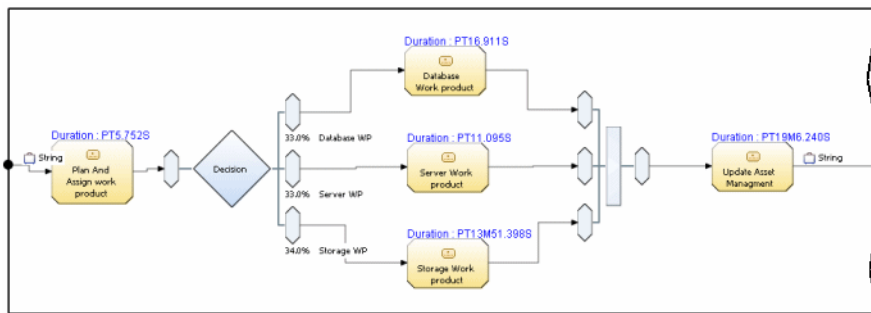


Figure 3-12 Diagram view

Dimensional view

The Dimensional view displays aggregated measures (sum, average, count, maximum, and minimum) across a subset of the process instances created (Figure 3-13). The criteria for that subset is determined by the dimensions selected.

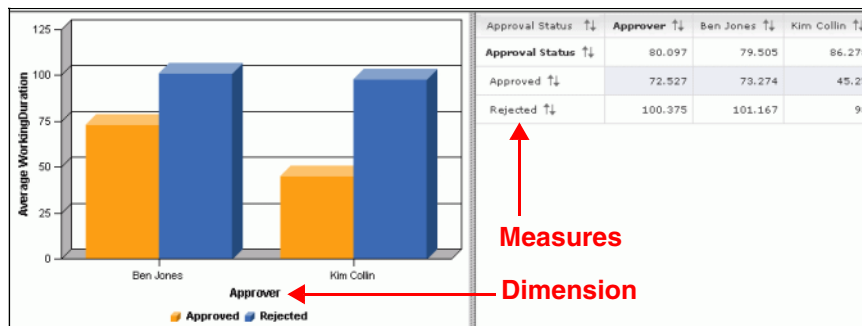


Figure 3-13 Dimensional view

Export Values view

The Export Values view is used to export actual values from the process so that this information can be imported to Modeler to close the loop (Figure 3-14). This feedback can enhance the modelling activities by providing real data from the field to the business analyst. Feedback values enables the business analyst to validate or amend the assumptions made when initially simulating the process in Modeler.

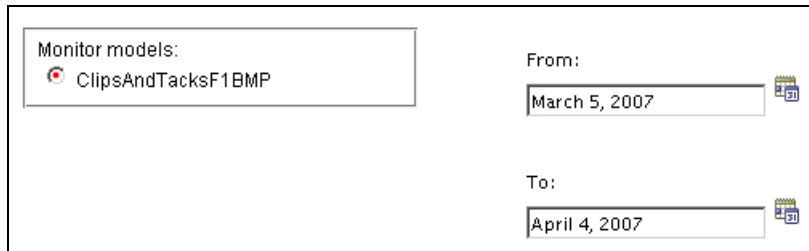


Figure 3-14 Export Actual Values view

Wiring different Monitor views

There are a set of Monitor view combinations where data displayed in a view can be based on data from another view.

The different combinations are:

- ▶ Instances view to Diagram view—This connection displays the metrics of a particular process instance in the diagram view.
- ▶ Alerts view to Instances view—This connection provides the ability to view the process instances that raised the alert.
- ▶ Organization view to Reports view—Values displayed in the report view are based on the user or the organization chosen from the organization view.

WebSphere Business Monitor topology

The standard topology of the Monitor is shown in Figure 3-15:

- ▶ The business process application with human tasks runs on one Process Server. It may have its own database. Users (customers) interface with the business process through Web front-end applications (for example). Events are generated, stored in the CEI database, and sent to the Monitor Server through a service integration bus (SIB).
- ▶ The Monitor Server runs on top of another Process Server. The Model Manager and the Adaptive Action Manager process the events. Data is stored

in the MONITOR database and processed using DB2 Cube Views. The state data in the MONITOR database is replicated to the DATAMART database.

- ▶ The Monitor Dashboard runs on WebSphere Portal Server. DB2 Alphablox uses the DATAMART database to display data in dashboard views. This is where all the business intelligence analysis is being performed. Business analysts interface with the Portal server to view the dashboard.
- ▶ An LDAP user registry is used for security purposes and for the Adaptive Action Manager. This registry could be installed on a separate server or in one of the three servers.

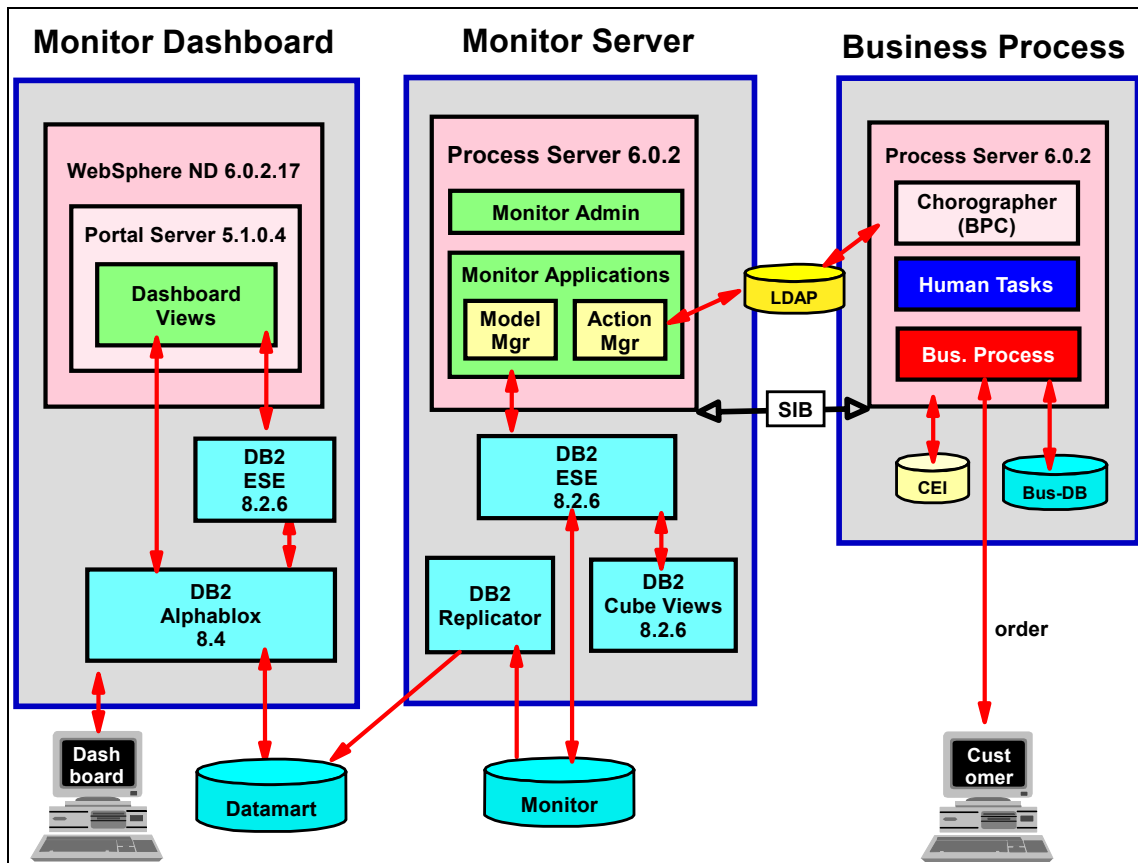


Figure 3-15 Monitor topology with separate servers

Such a configuration with two servers for the Monitor is recommended for high-performance or high-activity environments. Separating the Monitor Dashboard from the Monitor Server removes the high-load business analysis processing from the event handling and data propagation functions.

Simple Monitor topology

In a smaller (low activity) installation or for testing purposes, you can run the Monitor Server and the Monitor Dashboard on a single system (Figure 3-16).

As we progress in this book through our business process scenario, we use this simple topology where all the Monitor components are installed on a single system. The Monitor Server consumes events from a Process Server that runs on another system.

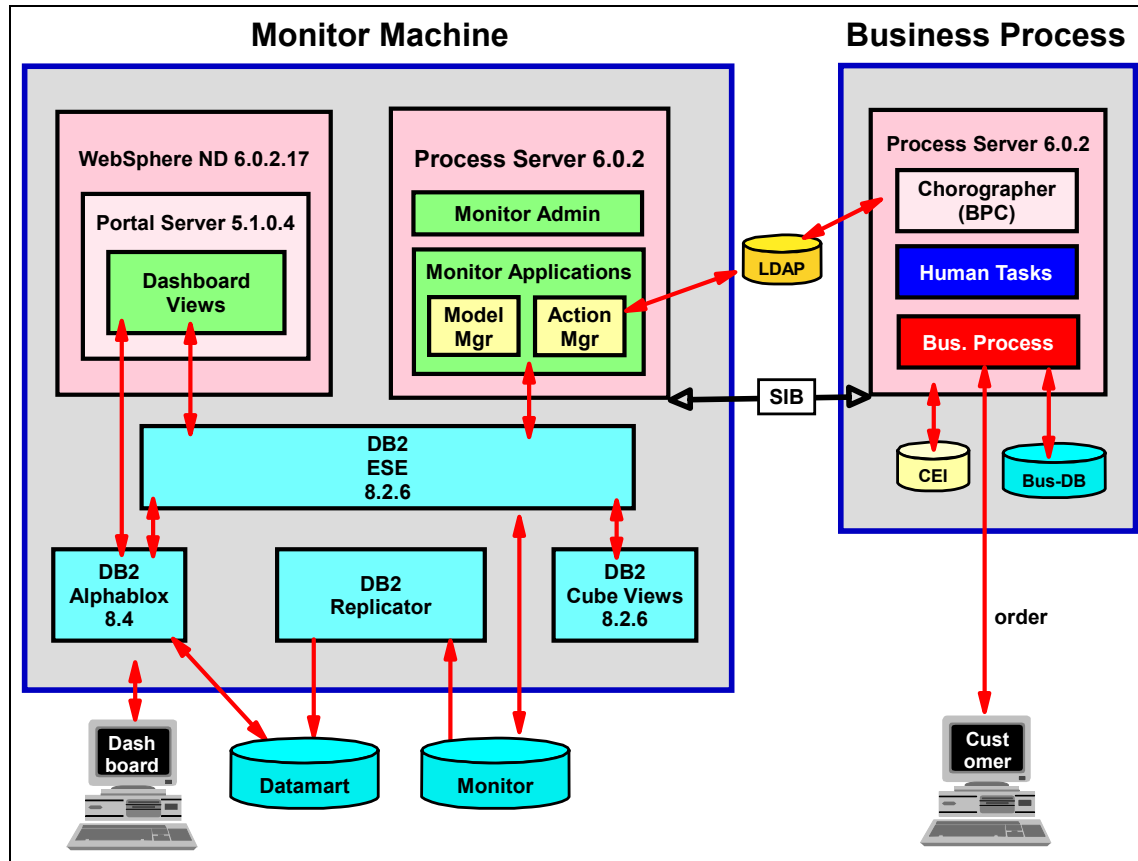


Figure 3-16 Simple project software configuration and topology

Note that running with such a simple configuration requires adequate main memory to run both the Monitor Server (Process Server) and the Monitor Dashboard (Portal Server) concurrently. The suggested minimum memory is 4 GB.

Single system topology

Version 6.0.2 of the products allows you to run the business process and the Monitor on the same Process Server, enabling you to run the application, the Monitor Server, and the Monitor Dashboard on one system (Figure 3-17).

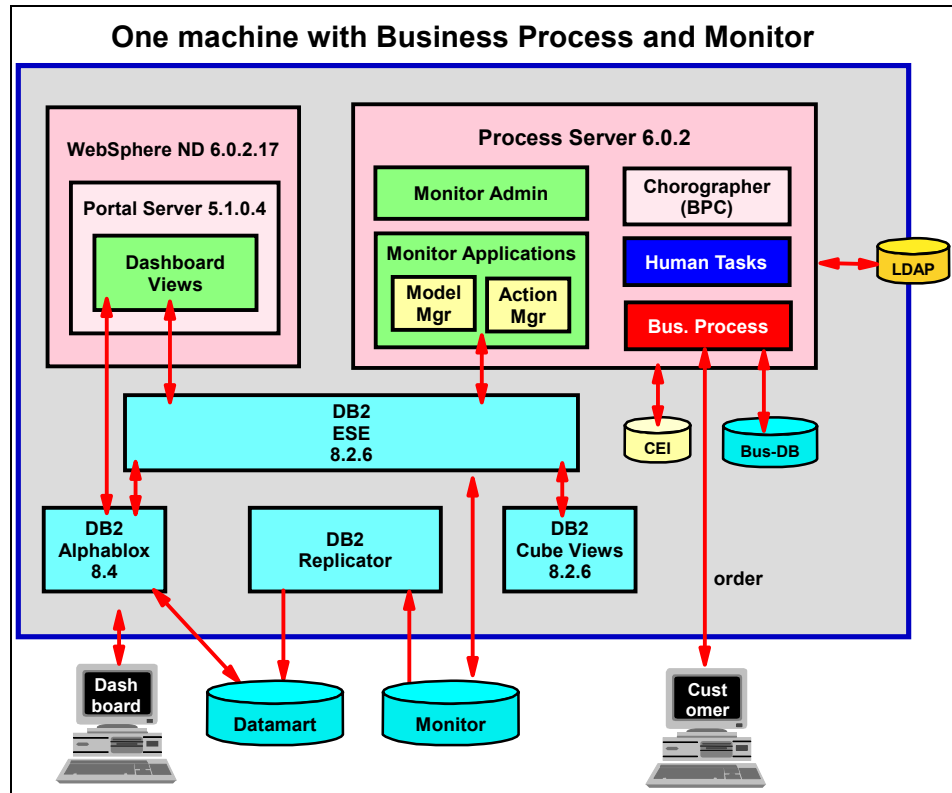


Figure 3-17 Single system topology

This topology is useful for testing, demonstrations, or low volume production.

Summary

This chapter described the architecture and internal and external components of WebSphere Monitor V6.0.2. We discussed the different scenarios where WebSphere Monitor can be used and the tools used to enable this. We also discussed the different WebSphere Monitor Dashboards provided by the product and the different capabilities that they offer.



Case study: ClipsAndTacks

This chapter describes a business process management (BPM) scenario, that shows how a business can use a full IBM WebSphere Business Integration solution to complete the following end-to-end business process lifecycle tasks:

- ▶ Model and simulate a business process.
- ▶ Define key performance indicators.
- ▶ Develop and test an application to implement the business process.
- ▶ Deploy and run the application on a server.
- ▶ Monitor the application to observe pre-determined key performance indicators.
- ▶ Import the observed data to make required revisions to the original process model in the Modeler.

The business scenario described in this document has been simplified in order to provide a full description of each stage of the BPM end-to-end process lifecycle. To avoid an overly large and unwieldy document, the authors' focus is on specific tasks, elements, and details, and not on presenting all possible facets of a complex business process. Each IBM software product included in the full BPM cycle has a large number of features and capabilities. For details on all of the available functions of these products, see the help documentation included with each product.

Case study: ClipsAndTacks Office Supplies Ltd.

This case study describes a fictional company that is seeking to improve one of its business processes. It shows how a business can quickly respond to the needs of its customers using an IBM business process management solution. The business described in this scenario, ClipsAndTacks Office Supplies Ltd. (abbreviated as *ClipsAndTacks* from now on) is experiencing a business problem that is negatively impacting its bottom line. The company needs to plan and implement a revised process that will address the business problem. To complete this business transformation, ClipsAndTacks will use the component products of the IBM business process management solution.

Background

ClipsAndTacks is a medium-sized office supply company operating in eastern Canada and the northeastern United States. The company has grown slowly and has achieved a significant customer base through its excellent customer service practices and reputation for quality products. Most ClipsAndTacks customers are businesses; ClipsAndTacks does not allow accounts for non-business customers.

Business problem

ClipsAndTacks has been losing customers to Office Market, its main competitor. Office Market is a national office supply chain which provides an online catalog and ordering process for its customers. From Office Market's Web portal, customers can view available products and submit an order 24 hours a day, 7 days a week.

Although it has been losing customers, and consequently revenue, ClipsAndTacks' costs have remained constant relative to the order handling process. They have maintained the same number of customer representatives to avoid introducing further delays into the ordering process due to longer call-waiting times. Each customer representative traditionally receives an increase in their hourly wage each year. The company also continues to maintain separate customer and product management systems, neither of which have been upgraded in several years. Both systems are prone to problems and outages, and require considerable maintenance.

Several customer surveys have indicated that ClipsAndTacks' customers are not satisfied with the ordering process. The telephone ordering procedure is time-consuming, and customers are frustrated at being placed on hold while waiting for the next available representative. Regular customers are frustrated at the amount of time it takes to receive their orders.

The delays are most often caused by the order review process, requiring each order over \$500 to be reviewed by an order manager for any credit risk before it is sent on to be filled. Customers repeatedly cite Office Market's online order process as a quicker and more convenient method for ordering their office supplies, and their comparatively quicker delivery time as a major advantage over ClipsAndTacks.

Summary of problems:

- ▶ Call center hours of operation are not convenient.
- ▶ The telephone order submission process is too long.
- ▶ The order review process delays shipments.
- ▶ Regular customers, in particular, resent delays due to order reviews.
- ▶ The company is losing customers and revenue.

Business objectives

As a result of the customer surveys, ClipsAndTacks' management has decided that the order handling process has to be updated so that it can fill orders in a shorter amount of time. Company management wants to establish an automated process that shortens order turnaround time, especially for trusted repeat customers.

The planned improvements include a new Web-based ordering system, which is a customer's access point to an almost totally automated ordering application. Simply put, it is an order handling application available 24 hours a day over the Internet.

The high-level business objectives of ClipsAndTacks are to increase revenue and reduce costs.

Note: We do not monitor and analyze costs in this document.

Specifically, management wants to achieve the following objectives:

- ▶ Reduce the average time from when orders are received to the time they are shipped to 3 days
- ▶ Achieve an order approval rate of 90% or better

Current order handling process

ClipsAndTacks publishes a product catalog that is mailed to its existing account customers. Customers can only place orders by telephone, with calls routed through the company call center. The call center is staffed by customer representatives and takes calls between the hours of 8:00 a.m. and 5:00 p.m. Eastern Time, Monday to Friday.

For new customers, a customer service representative manually enters the contact and address information to the customer database and assigns a customer number. Existing customers have to provide their customer number so that the representative can retrieve the customer record containing the customer's information. When the customer information is retrieved, the customer provides the details of the order.

All orders are forwarded to an order manager. If an order total is \$500 or more, the order manager must review it before sending it to the warehouse to be filled. The order manager assesses the credit risk of each order: if the order is deemed an acceptable risk, it is sent on to the warehouse to be filled; if the order is not deemed an acceptable risk, the order manager cancels the order and sends an e-mail notification to the customer. All approved orders are sent to the warehouse to be filled.

Summary of current business rules:

- ▶ Customers can only order by telephone.
- ▶ Orders are accepted only from 8:00 a.m. to 5:00 p.m. Eastern Time, Monday to Friday.
- ▶ Customer service representatives handle all inbound order requests.
- ▶ All orders are forwarded to the order manager for review.
- ▶ Orders over \$500 must be approved by the order manager.

Planned revisions to the order handling process

ClipsAndTacks' management wants to implement an order submission process that will allow ClipsAndTacks to compete with Office Market. In response to a common complaint in the customer surveys, the new process will eliminate the need for contact between customers and customer service representatives when an order is placed. Customers will be able to browse the ClipsAndTacks product catalog and enter their own order information using a Web application. New customers will be able to enter their company information and receive a customer account number immediately.

Customers who have a customer number will be able to enter it and prompt the Web application to retrieve their information and pre-fill the Web form with their address and preferred shipping information. The new Web application, including the product catalog and order form, will be available 24 hours a day, 7 days a week.

When the customer submits the order, a business rules engine will be checked to ensure appropriate action on the order. In response to comments from regular customers, the threshold for order review will be raised to \$750. If an order is for an amount under \$750, it is approved automatically, pending a check of the account status (that is, are there any outstanding charges against the account that have not been paid). If the account is in good standing, the order is sent for shipping. If the account is not in good standing, it is sent to an order manager for review. If an order is for an amount greater than \$750, it is sent to an order manager for review. Based on the review, the order manager decides whether to send the order for fulfilment or to cancel the order and notify the customer.

Summary of revisions:

- ▶ Customers can order online (shortens the order process).
- ▶ Orders are accepted 24 hours a day, 7 days a week.
- ▶ Implementation of rules/policy engine is now included.
- ▶ The threshold for order review has been raised to \$750.
- ▶ Orders over \$750 must be approved by the order manager (shortens average order time and increases percentage of approved orders).

Note: We must reiterate that we are presenting a simplified process. The outright cancellation of an order is less likely than a request for prepayment, or some other arrangement between the companies. A simplified scenario is presented in order to show details of each phase of the full cycle.

Key performance indicators

Key performance indicators (KPI) are the detailed specifications required to track business objectives. Each KPI is associated with a specific process, and is quantifiable, measurable, and results-oriented.

The ClipsAndTacks management team wants to be able to measure the results of the revised process when it is implemented to ensure that it is helping to meet the company's business objectives. To measure the revised order handling process, management has identified two key performance indicators that will measure the success of the new process. Each of these key performance indicators will comprise at least one metric.

To gauge the performance of the revised process and determine if it is helping to achieve the company's business goals, the ClipsAndTacks management team will set the following KPIs:

▶ **Average order fulfilment time is 3 days or less:**

- Target: 3 days
- Lower margin: 1 day
- Upper margin: 5 days

This KPI is based on measuring the average processing time for shipped orders.

▶ **Number of approved orders is at least 90%**

- Target: 90%
- Lower target margin (%): 85%
- Upper target margin (%): 100%

This KPI is based on measuring the number of shipped orders versus the total number of orders.

Roles

The following key roles take part in the ClipsAndTacks BPM scenario:

▶ **Customer**

Typically, ClipsAndTacks customers are business professionals who purchases office supplies for a company. Customers are comfortable with Web processes, and expect an order process to be quick and simple. They also expect that orders will be delivered promptly.

▶ **Business analyst**

ClipsAndTacks employs a business analyst on staff. The business analyst is responsible for understanding the company's existing processes and designing ways to improve those processes. The analyst gathers and documents information about the company's processes by reviewing reports, interviewing employees, and observing performance. The analyst then models and simulates current and planned practices.

The business analyst also defines the business measures, such as the key performance indicators, that will be monitored. In our scenario the business analyst uses WebSphere Business Modeler to model, simulate, and analyze the business process.

▶ **Order manager (order approver)**

A ClipsAndTacks staff member who has the authority to accept or reject a purchase order. Orders that are over the limit of \$750 are routed to the order manager, who either approves or declines the order. ClipsAndTacks management provides guidelines to the order manager that help in making the decision.

▶ **System architect (integration developer)**

The system architect is responsible for designing, building, and testing the implementation of the revised process that has been modeled by the business analyst. In our scenario the architect uses WebSphere Integration Developer to implement the business process and WebSphere Process Server to test the implementation.

▶ **Application programmer**

The application programmer is responsible for implementing some of the specific services that the systems architect has specified at a higher level. For example, the programmer writes the Web application to be used by the customers, and the Java code to interface with external services.

▶ **System administrator**

The system administrator is responsible for deploying the finished application in a production environment using WebSphere Process Server and for monitoring the runtime process using WebSphere Business Monitor. The administrator ensures that the component programs operate as they are designed. The administrator records relevant data about the process and produces performance reports for company management.

Throughout this book, key roles are described for each stage in the process.

Subsystems

Figure 4-1 shows the abstract logical subsystems that comprise the solution architecture. The essential subsystems for the ordering process are shown in green with a bold border.

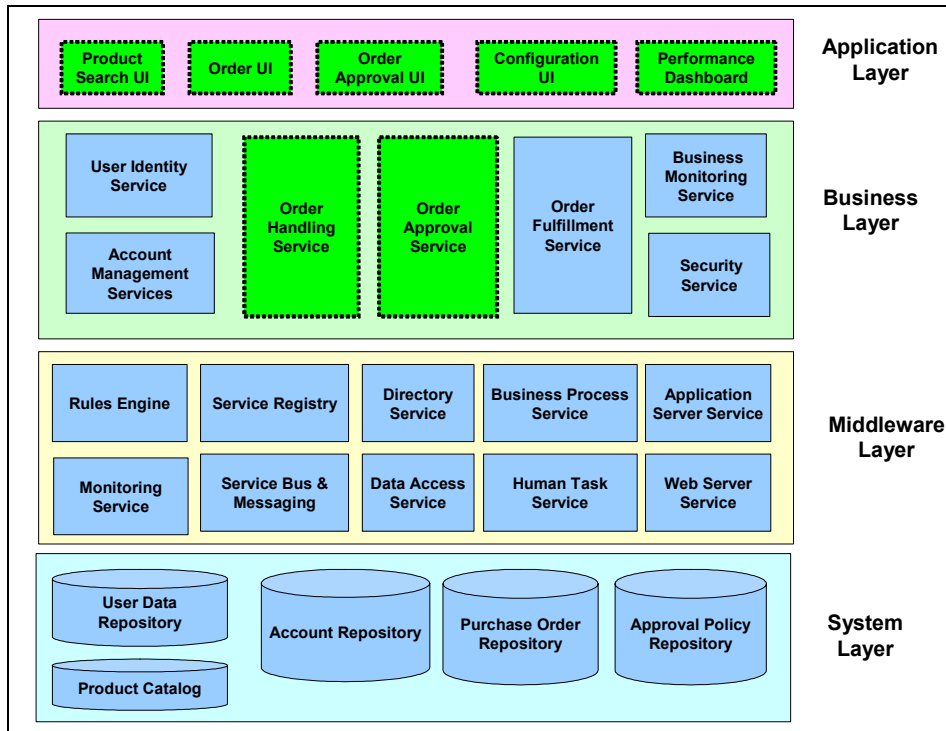


Figure 4-1 Solution architecture

Table 4-1 describes the essential subsystems.

Table 4-1 Abstract logical subsystems that comprise the solution architecture

Subsystem	Description
Product Search UI	Web-based user interface for customers to search and/or select products from an integrated product catalog.
Order UI	Web-based user interface for customers to enter order information, such as quantity and shipping method, or to view the details of existing orders.
Order Approval UI	User interface for the order manager to view and authorize orders.
Configuration UI	User interface for ClipsAndTacks' staff to configure the ordering process, specifically, the order approval process.

Subsystem	Description
Performance Dashboard	Web-based user interface for Business Performance Analyst to query and view business performance results
Order Handling Service	A business process that responds to the Order UI and creates new orders or displays existing order status. The process determines whether an order can be automatically processed according to the procurement policies.
Order Approval Service	A workflow process that is used by the order approval staff to validate and approve orders.

Hardware prerequisites

You must have installed and configured the full BPM suite of products before completing the set of tasks described in this book. See the individual product documentation for installation and configuration instructions. System hardware requirements are shown in Table 4-2.

Table 4-2 Hardware requirements

WebSphere Business Modeler	
Processor	Pentium® III 500 MHz (or equivalent) or faster
RAM	1 GB (2 GB recommended)
Disk Space	900 MB to install
Display	Minimum 1024 by 768 resolution (1280 by 1024 or more is recommended)
WebSphere Integration Developer (with Monitor Development Toolkit)	
Processor	Pentium 2 GHz (or equivalent) or faster
RAM	Minimum 2 GB (minimum 1 GB without Monitor Development Toolkit)
Disk Space	5.8 GB to install (including 1.15 GB temporary disk space)
Display	Minimum 1024 by 768 resolution (1280 by 1024 or more is recommended)
WebSphere Process Server	
Processor	Pentium 1 GHz (or equivalent) or faster
RAM	1 GB (2GB recommended)

Disk Space	2 GB to install
Display	Minimum 1024 by 768 resolution
WebSphere Business Monitor (all components on one system)	
Processor	Pentium 2 GHz (or equivalent) or faster
RAM	4 GB (2 GB each for separate Monitor Server and Dashboard Server)
Disk Space	5 GB to install (Monitor Server, Portal Server, DB2 + Cubes + Alphablox + LDAP)
Display	Minimum 1024 by 768 resolution

More information

For more information about system requirements, refer to these Web sites:

- ▶ WebSphere Business Modeler:
<http://www.ibm.com/software/integration/wbimodeler/advanced/sysreq/>
- ▶ WebSphere Integration Developer:
<http://www.ibm.com/software/integration/wid/sysreqs/>
- ▶ WebSphere Process Server:
<http://www.ibm.com/software/integration/wps/sysreqs/>
- ▶ WebSphere Business Monitor:
<http://www.ibm.com/software/integration/wbimonitor/requirements/>

Summary

In this chapter we introduced the ClipsAndTacks company. We will use their order handling business process to demonstrate how the IBM business process management products can be used to model, implement, run, and monitor the ClipsAndTacks application.



Part 2

Modeling the business

In Part 2 we describe the modeling activities to be carried out by the ClipsAndTacks business analyst.

We start with the model of the current business, *Order Handling (Current)*, and simulate its behavior to find out the bottleneck. Then we describe how to improve the model to provide for a better customer response time and for more orders approved. This model is referred to as *Order Handling (Future 1)*. We simulate and analyze the new model.

Finally, we decide what business measures to define and what key performance indicators (KPI) to measure when we implement the new model in WebSphere Process Server and monitor the execution using WebSphere Business Monitor.



Modeling the current business process

This chapter describes how the ClipsAndTacks order handling process, *Order Handling (Current)*, was modeled and imported into WebSphere Business Modeler.

The key to a successful re-engineering of a business process is to thoroughly understand the details of the existing process and to accurately forecast the results of the changes to that process. In order to analyze and predict the outcome of a revised process, a business first has to create an accurate representation of the process with a model and then must study how that process performs under different conditions.

A common error that businesses often make when they set out to revise their processes is failing to fully investigate and understand their current process and their objectives in re-engineering that process. Without devoting the necessary time to study, analyze, and plan, many businesses find that either they have not adequately addressed the original problem, or they have simply exchanged a recognized problem for another unanticipated one.

Introduction to modeling the current business process

Before implementing a revised order handling process, the company must comprehensively document, model, and analyze the current process so that it can fully understand where the business problems lie and set realistic targets for the process improvement.

At ClipsAndTacks, the lead business analyst has been given the task of designing the revised order handling process. On its surface, this step seems fairly straightforward. To ensure that his model is as accurate as possible, the business analyst must first document and sketch the current process flow, and list all of the key resources (people, equipment, material), business items (documents, records, products), and business rules (decision logic) included in the process.

Note: For detailed instructions on how to create each of the elements described in following sections, refer to the help documentation included with WebSphere Business Modeler.

Documenting the current process

For the remaining of this chapter, you are now the business analyst.

The first step is to investigate and fully document the current process. By reviewing available data reports and interviewing each employee involved in the process, you can gain an understanding of how the process currently works, and where the problem areas lie. For ClipsAndTacks, you must monitor each stage of the process, observing the employees' interaction with customers, and the progress of orders through the complete order handling process.

You have to focus on recording all key aspects of the process:

- ▶ What is the process flow?
- ▶ What are the resources used in the process?
- ▶ What are the business items acted on by the process?

As you study the process and gather your data, record the following items:

- ▶ Activity inputs and outputs
- ▶ Task variations and when those variations occur
- ▶ Alternative tasks
- ▶ Complete task descriptions
- ▶ Roles associated with tasks

After you have gathered sufficient data, record the findings and map out a rough flow diagram of the current process. Make note of the output at each stage of the process, indicating stages where there is more than one possible outcome. List all of the activities, resources, roles, and business items involved in the process.

Process flow

When you have gathered all available data and observed the current process in action, you will be able to write out the process flow, including any spot where there is more than one possible outcome. At ClipsAndTacks, the current order handling process flow takes place as follows:

- 1. The customer service representative receives the call from the customer.**
- 2. First determine if the customer has an account.**
 - 2.1 If the customer has an existing account:**
 - Enter the account number.
 - Enter the order information.
 - Approve order or send for review.
 - 2.2. If the customer does not have an account:**
 - Enter customer information and assign an account number.
 - Enter the order information.
 - Approve order or send for review.
- 3. If the order is approved:**
 - Send the order to the warehouse.
 - Record order in order records database.
 - Issue packing slip.
 - Ship the product.
- 4. If the order is sent for review:**
 - Review the order manually.
 - Determine if the order is an acceptable credit risk.
 - 4.1 If the order is an acceptable credit risk:**
 - Send the order to the warehouse.
 - Record order in order records database.
 - Issue packing slip.
 - Ship the product.
 - 4.2 If the order is not an acceptable credit risk:**
 - Cancel the order.
 - Send an order cancellation notification to a customer.

At this stage, you have enough information to sketch out the process. You can use WebSphere Business Modeler to sketch out the rough process, or simply draw it using pencil and paper. The process flow for the Order Handling process might look as shown in Figure 5-1.

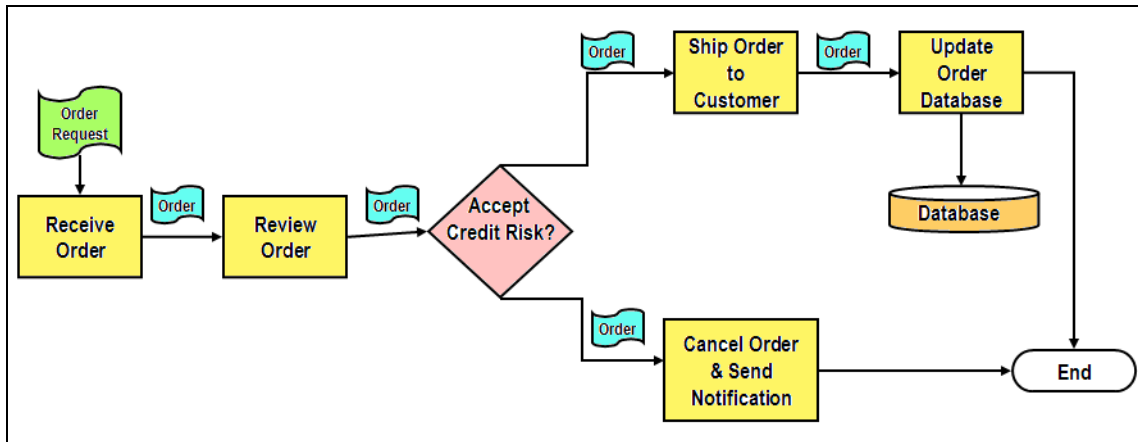


Figure 5-1 Manual process flow draft

Business items

Review the flow to determine the business items that are acted on during the process. Remember that business items are business documents, work products, or commodities that are transformed in business operations. You can model as a business item anything that is created, assembled, inspected, tested, modified, or worked upon. Business items undergo changes as they are passed from one process step to the next.

For the ClipsAndTacks order handling items, the following business items are acted on in the current order handling process:

- ▶ Request (a verbal request from the customer)
- ▶ Order (a list of products requested by the customer with the shipping and billing information)
- ▶ Customer record
- ▶ Packing slip with packaged product
- ▶ Cancellation notification to a customer

Resources

A key to documenting any process is determining the resources and roles required to complete each of the activities. Resources represent the people, equipment, or material used to perform a project or a task. Resources are not the same as business items. The objects that undergo changes and are passed from one process step to the next should be modeled as business items, whereas the things that are performing the work or are required prerequisites for this work, such as machines, fuel, vehicles, or skilled personnel, should be modeled as resources.

Resources are represented by the following people roles and subsystems in the ClipsAndTacks order handling process:

- ▶ Customer service representative
- ▶ Order manager
- ▶ Shipper
- ▶ Customer records system
- ▶ Product catalog system
- ▶ Order management system

Timetables

An accurate model must also take into account the schedules of the resources involved in the process. In WebSphere Business Modeler, you can define timetables that specify what times certain resources are available. The following timetables are required to indicate the work hours of the key roles in the current order handling process:

- ▶ Day shift
- ▶ Weekend

Creating the current process diagram using the Modeler

In this section we describe how to create the model of the current process flow.

Detailed steps to create the model

Business modeling is an iterative process, requiring the business analyst to continually revise the process as they gain a deeper understanding of the goals, requirements, and individual activities involved. The business analyst must continue to meet with subject-matter experts to gather information and validate draft models.

The current order handling process diagram reflects the current process flow as you have documented it.

Because this book is not focused on just the modeling of the process, but instead it has the goal of documenting the complete end-to-end business process management lifecycle, we will not go through each step of creating a current process model in this chapter.

The detailed documentation on how to use Modeler is available in the help documentation included with the product. In Chapter 7, “Modeling the Future 1 business process” on page 121, we will go through the detailed steps for modifying an existing process model. In this chapter we import the current order handling process model that was previously built by the business analyst.

Modeler quick guide

This book is not a complete guide to WebSphere Business Modeler V6. However, we provide here a few simple guidelines for working with the Modeler.

Start the Modeler using *Start* → *Programs* → *WebSphere Business Modeler* → *WebSphere Business Modeler*.

Perspectives

Most often, you work in the Business Modeling perspective. You can also use the Resource perspective to see the underlying physical files.

Modeling mode

The Modeler supports a number of modes that provide more details and capabilities:

- ▶ Basic—Focuses on purely business tasks
- ▶ Intermediate—Adds input and output details and formal expressions
- ▶ Advanced—Models that form the basis of software applications
- ▶ WebSphere Business Integration Server Foundation
- ▶ WebSphere MQ Workflow
- ▶ WebSphere Process Server


The first three modes are for modeling the processes, business items, resources, organization, and so forth. The last three modes enable you to define details that will be used when the processes are exported for usage in one of the three products.

For our scenario, we work using the *Advanced* mode for modeling, and the *WebSphere Process Server* mode before exporting to Integration Developer and process Server. The mode is set by selecting *Modeling* → *Mode* → *Advanced* (for example).

Help and tutorial

The Modeler provides extensive help. Select *Help* → *Help Contents* to open the help facility. If you are a beginner, expand *WebSphere Business Modeler Advanced* and select *Samples and tutorials* → *Tutorial: Quickstart*. Then go through the complete tutorial to become familiar with the Modeler.

Screen layout

Click the *Apply 4-pane layout* icon  to see the Project Tree, and edit area, the Outline, and Attributes/Errors. This layout gives you access to all important tasks.

Importing the current process model using the Modeler

Listed below are the detailed steps for the importing of the existing process model that was previously built using the Modeler product.

You can import models or definitions from several different formats into the WebSphere Business Modeler Version 6.

Specifically, you can import files in the following formats:

- ▶ WebSphere Business Modeler project (.mar, .zip)
- ▶ WebSphere MQ Workflow (.fdl)
- ▶ WebSphere Business Integration Workbench 4.2.4 (.org)
- ▶ Delimited text (.csv, .txt)
- ▶ Microsoft Visio (.vdx)
- ▶ WebSphere Business Modeler XML (.xml)
- ▶ Monitoring result (.xml)
- ▶ Business services and service objects (.wsdl, .xsd)

We import the current order handling process model that was previously modeled and exported as a WebSphere Business Modeler project zip file.

The Business Integration Modeler project MAR file (Modeler ARchive) can be found in the sample code available with this book (see Appendix C, “Additional material” on page 647):

```
SG247148\sampcode\model\ClipsAndTacks Current.mar
```

To import the MAR file, complete the following steps:

- ▶ In the Project Tree, right-click and select *Import*. The Import wizard appears.
- ▶ Select *WebSphere Business Modeler project* and click *Next* (Figure 5-2).

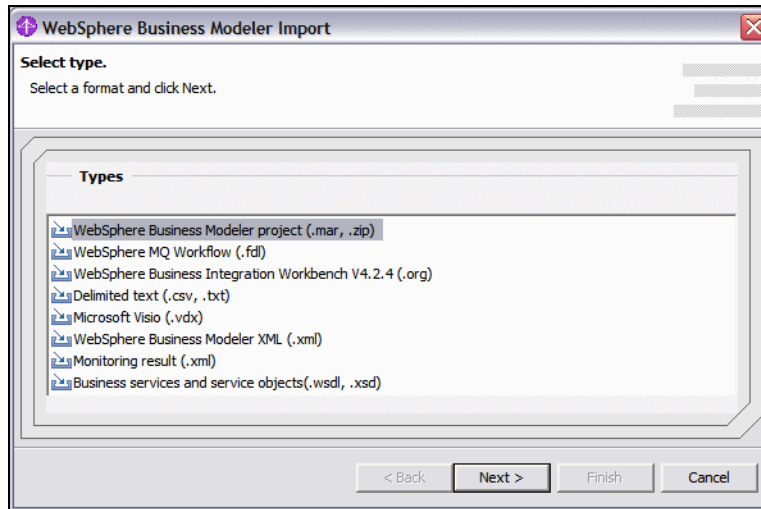


Figure 5-2 Select model import type

- ▶ Click *Browse* to select the source directory that contains the file you want to import (Figure 5-3).

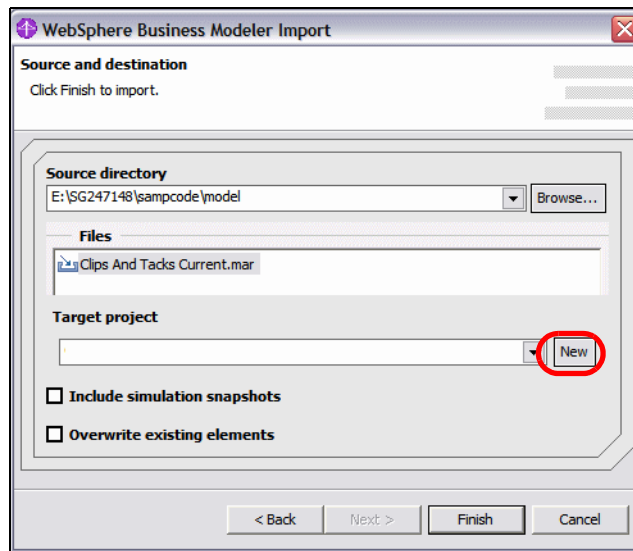


Figure 5-3 Importing a model

- ▶ In the Files list, select the MAR file containing the project or element that you want to import. Select the `Clips And Tacks Current.mar` file from:

`SG247148\sampcode\model`

- ▶ Clear *Include simulation snapshots* (there are no simulations).
- ▶ Optionally select *Overwrite existing elements* to replace any of the elements with the same name as an element being imported. Otherwise, you will be warned when an element is about to be overwritten.

Note: If you choose to overwrite a catalog, be aware that the entire catalog and all of its contents will be deleted and the contents of the new catalog will be imported instead.

- ▶ In the Target project field, select an existing project from the drop-down list or click *New* to create a new project **Clips And Tacks Current**. Click *Finish*. (Figure 5-4).

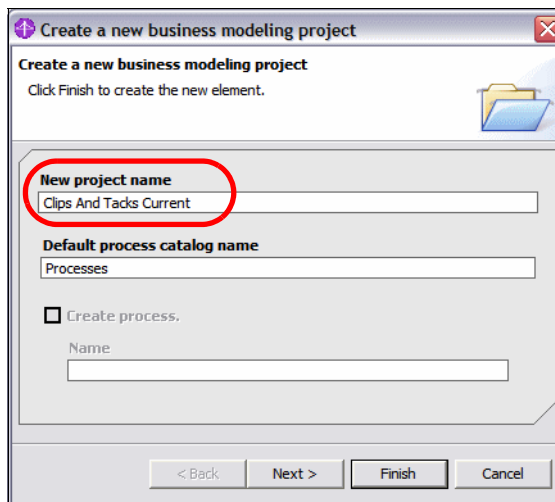


Figure 5-4 Creating a new project *ClipsAndTacks*

- ▶ The project is created and its name is filled into the dialog. Click *Finish* to import the MAR file.
- ▶ If there are any errors or warnings during the import process, click *Details* to read them.

The project files have been imported into the project you specified. The Project Tree view of the project is shown in Figure 5-5.

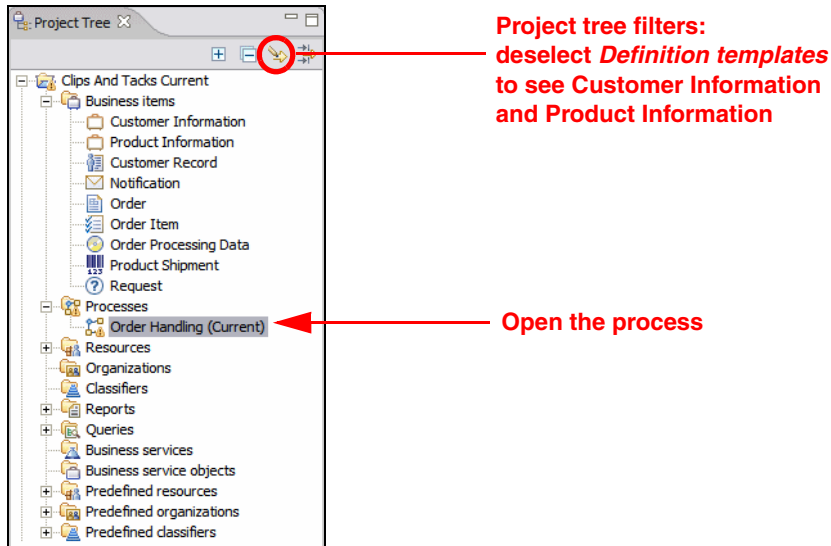


Figure 5-5 Project Tree of the current order handling process

Order Handling (Current) process

The imported current order handling process model is shown in Figure 5-6.

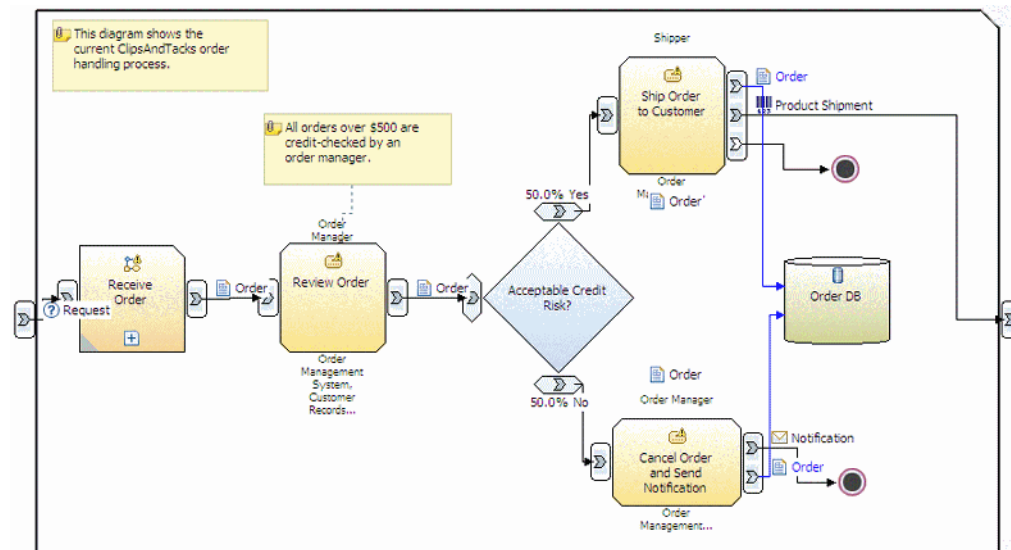


Figure 5-6 Current order handling process model (compressed to fit)

These are the activities in the order handling process:

- ▶ **Receive Order**—A local process that handles the interaction of the customer with an agent.
- ▶ **Review Order**—An order manager reviews all orders over \$500 and decides if the order should be approved or declined.
- ▶ **Ship Order to Customer**—If the credit risk is acceptable, the order is processed, recorded in an order database, and shipped.
- ▶ **Cancel Order and Send Notification**—If the order is declined, a cancellation notification is sent to the customer.

The last three activities are so-called *human tasks*, performed by a person. The input to the process is a customer request, and the output is the product shipment. The default probability for each branch of Acceptable Credit Risk is 50%.

Embedded process

To view the embedded process, Receive Order, select the process and *Edit*. The diagram of the Receive Order process is shown in Figure 5-7.

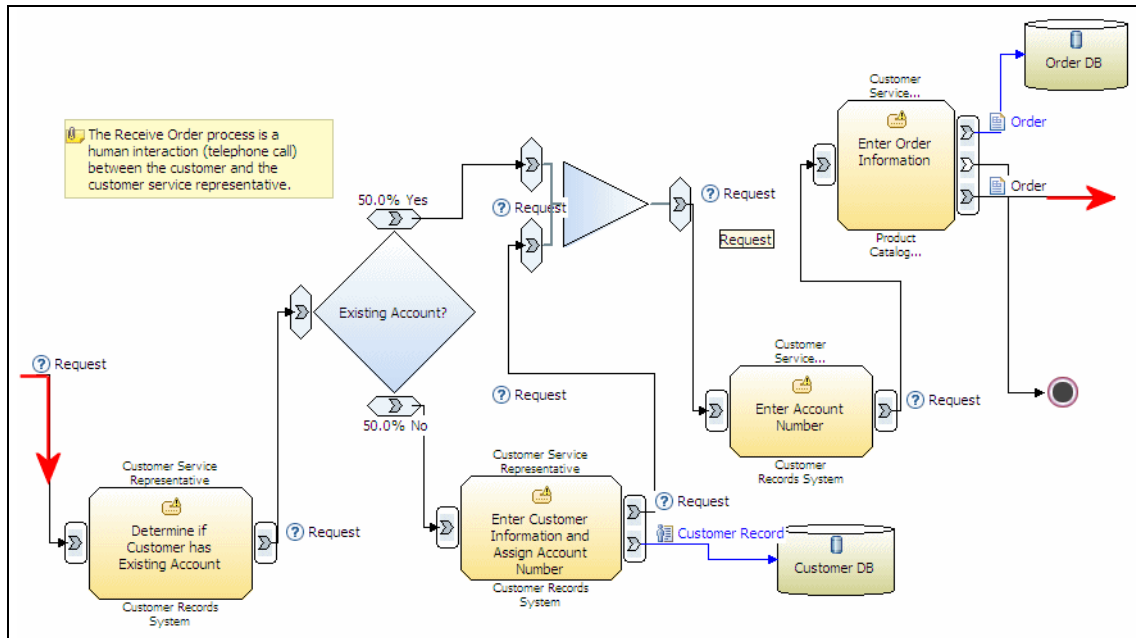


Figure 5-7 Receive Order subprocess (compressed)

To go back to the order handling process, right-click in an empty space in the diagram and select *Parent*.

Business items

The business items used during the process are listed here:

- ▶ **Order**—The Order is the main business item that flows through the whole process (Figure 5-8). An order consists of an OrderNumber, an OrderStatus, the TotalPrice of the order, a Customer with all its information, the OrderItems (which products), and a processing field (automaticApproval coming from **Order Processing Data**).

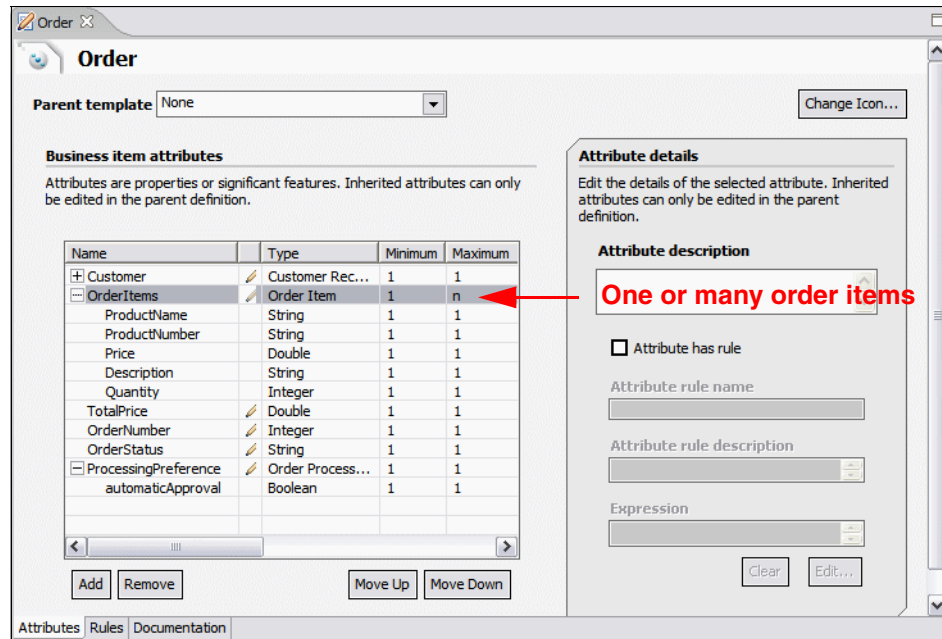


Figure 5-8 Business item: Order

- ▶ **Order Item**—An order item consists of a Quantity of a product ordered. The product attributes come from **Product Information** (Figure 5-9).

Name	Type	Minimum
ProductName	String	1
ProductNumber	String	1
Price	Double	1
Description	String	1
Quantity	Integer	1

Product Information

Figure 5-9 Business item: Order Item and Product Information

- ▶ **Customer Record**—The data kept for a customer is composed of basic data coming from **Customer Information** and additional attributes, Rating, and AvailableCredit (Figure 5-10).

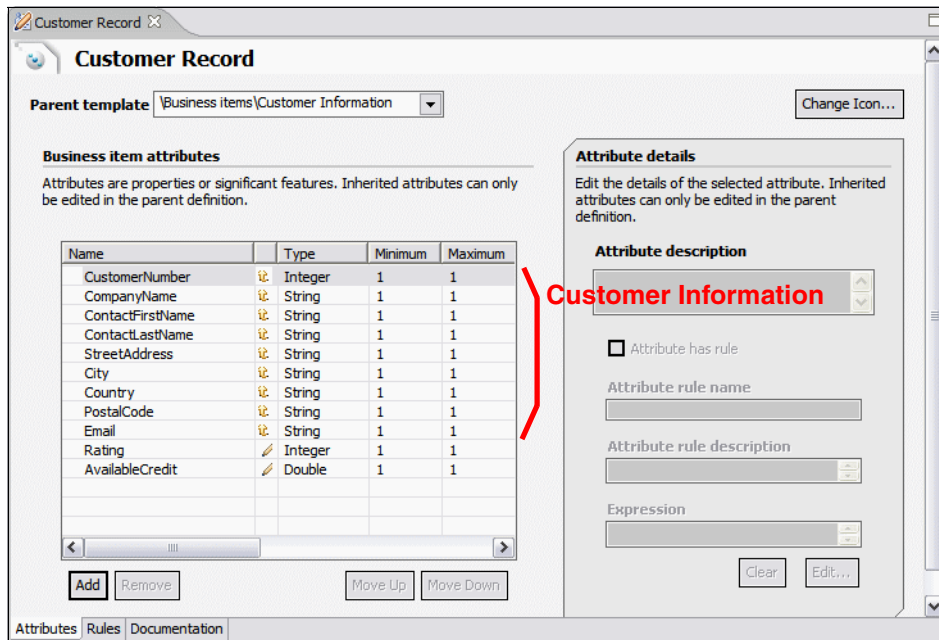


Figure 5-10 Business Item: Customer Record

- ▶ **Product Shipment**—Similar to the order, consist of a Packing Slip Number, OrderNumber, TotalPrice, Customer, and OrderItems.
- ▶ **Notification**—Consists of the e-mail address and the text to be sent to a customer when an order is declined.
- ▶ **Request**—This is the request coming from a customer to order some products.

The business items Customer Record, Order, and Order Item will be used for the database design when we implement the application in a WebSphere server.

Description of some business item attributes

Most of the attributes are self-explanatory, the others are explained here:

- ▶ **OrderStatus**—The status of an order during the process: NEW, APPROVED, DECLINED, SHIPPED.
- ▶ **automaticApproval**—An order can be automatically approved if the total price is below a certain value. This attribute is either false or true.
- ▶ **Rating**—The credit rating of a customer as retrieved from an external service. This value will be used by the order manager to decide if an order should be approved or declined.

- ▶ `AvailableCredit`—The available credit of a customer. This value will be used by the order manager to decide if an order should be approved or declined.

Resources

These resources are defined for our scenario:

- ▶ Customer service representative—Takes the phone call from a customer.
- ▶ Order manager—Decides if an order should be approved or declined.
- ▶ Shipper—Packages the products and ships to the customer.
- ▶ Customer Records System, Order Management System, Product Catalog System—Computer applications used for customer information, orders, and the product catalog.
- ▶ Day Shift and Weekend—Time tables.

Organizations and classifiers

No organizations and classifiers have been defined for our scenario.

Summary

In this chapter we described how ClipsAndTacks modeled their current business to get a better understanding of the business process, the activities, and to prepare for simulation and analysis.



Simulating and analyzing the current process

This chapter describes how the ClipsAndTacks current process is simulated and analyzed in the Modeler.

The key to a successful simulation and analyzing of the current process is to describe correctly what information we should gather from the real world business process to simulate a process.

The first part of this chapter provides an example of a set of information required to implement a representative simulation and analysis.

The second part of this chapter shows a demonstration on how to gather and analyze information provided by the simulator of WebSphere Business Modeler.

Overview of process simulation

These are the major steps to run a simulation of a process:

- ▶ Define resources and probabilities:
 - Corporate strategies
 - Process flow
 - Human resources needs and costs matrix
 - Duration matrix
 - Human resource availability matrix
 - Probabilities on decision matrix
 - Probabilities on output matrix
- ▶ Define a simulation profile and attributes related to the simulation runs
- ▶ Enter all simulation attributes in the Modeler
- ▶ Run a simulation snapshot
- ▶ Analyze simulation results

ClipsAndTacks process assessment for the simulation

After modeling the business process, you can use WebSphere Business Modeler to simulate the running of the process. Simulating allows you to assess the performance of the process, generate statistics about its execution, and pinpoint potential areas of improvement. A process simulation is a simulated performance of a real world business process in a virtual environment.

Before simulating the current process, the lead business analyst in charge of this project must organize workshops with operational officers to gather the following information about the real world business process:

- ▶ The corporate strategy, also called business objectives, is described in Chapter 4, “Case study: ClipsAndTacks” on page 65.
- ▶ The documentation and the design of the processes are described Chapter 5, “Modeling the current business process” on page 77.

As a result of these workshops a number of documents are produced.

Role resources matrix

The role resources matrix (Figure 6-1) shows the number of people for a specific role and resources assigned to activities. This matrix also shows the cost by roles. Usually the cost is defined by the salary divided by the unit of measure, an hour in our case.

Resources / Activity	Costs USD per hour	Determine if Customer has Existing Account	Enter Customer Information and Assign Account Number	Enter Account Number	Enter Order Information	Review Order	Cancel Order and Send Notification	Ship Order to Customer
Customer Service Representative	11.00	1	1	1	1			
Order Manager	20.00					1	1	
Shipper	10.00							1
Order Management System	0.00					1	1	1
Customer Records System	0.00	1	1	1		1		

Figure 6-1 Roles matrix: Cost and resource/role per activity

Duration matrix

The duration matrix (Figure 6-2) shows the duration of human tasks for a specific role and a specific activity. In this example, there is only one role per activity, but there could be multiple human roles for one activity.

Note: The total duration is not equal to the sum of the resources durations, because some resources are used in parallel.

	Determine if Customer has Existing Account	Enter Customer Information and Assign Account Number	Enter Account Number	Enter Order Information	Review Order	Cancel Order and Send Notification	Ship Order to Customer
Activity duration	20 sec	6 min	20 sec	12 min	20 min	2 min	16 min
Resources / Activity							
Customer Service Representative	20 sec	5 min 45 sec	10 sec	12 min			
Order Manager					20 min	2 min	
Shipper							15 min
Order Management System					1 min	2 min	1 min
Customer Records System	10 sec	15 sec	5 sec		1 min		

Figure 6-2 Duration matrix: Duration of activity by role and resources

Availability matrix

The human resources availability matrix (Figure 6-3) shows the timetables assigned to human roles. In our business case, only one timetable named *Day Shift* is used (described as a resource in the Modeler).

The day shift is defined as:

- ▶ 9 working hours a day
- ▶ Working days are Monday to Friday
- ▶ Working hours 8:00 AM to 5:00 PM






Timetable / Resources	Customer Service Representative	Order Manager	Shipper	Order Management System	Customer Records System
Day Shift					

Figure 6-3 Availability matrix: Time table per role

Note: The Weekend timetable is used as an exemption for the day shift timetable.

Decision probabilities

One of the process model components is a decision. You can assign a probability on decision choices (for example: is a customer an existing customer?), which determines the method of selecting a path through the process (Figure 6-4).

Decision / Probability	Yes	No
Acceptable Risk Credit	70%	30%
Existing Account	50%	50%

Figure 6-4 Decision matrix: Probability (yes/no) per decision

Note: WebSphere Business Modeler allows us to define probabilities on output criteria, which are the allowable combinations of outputs for an activity. In our case, you could find an example of multiple outputs with a probability defined in the activity Ship Order to Customer.

Simulation profile information

The assessment of the current process provides information about the duration of the benchmark to reflect a representative simulation. For our scenario, we define:

- ▶ The number of tokens for the simulation: **270**
 - A token represents a unit of work that is received by a process and transferred between different activities in the process flow. By specifying token creation settings, you define the quantity and rate of inputs that the process handles in a simulation run. For our example, a token represents a request. Assuming that we get 30 requests per hour, there will be 270 requests per 9 hour day.
- ▶ The maximum duration of the benchmark: **365 days**
 - Specify the maximum duration that a simulation will run. The maximum duration is the real time during which the simulation occurs.
- ▶ The start date and time of the simulation: **Monday, October 24, 2005 08:00:00 AM**
 - Specify start and end dates and times to define the virtual time in which the simulation takes place.
- ▶ The time measurement unit for results: **Minutes**
 - Select a unit of time that will be used for defining time-related distributions and for recording the results of process simulations.
- ▶ The distribution model for requests: **uniform distribution by minutes**
 - Measurements using any variable, even the same variable on the same subject, result in different outcomes. The pattern of different outcomes is called the distribution, which can be described mathematically and graphically. The distribution describes the relative number of times each possible outcome will occur in a number of trials.

Note: For more information about distribution models, refer to the product documentation: *Simulation processes* → *Setting Simulation* → *Attributes* → *Specifying token creation settings*

- ▶ Steady delay for the process: **0 minute**
 - Specify a period that must elapse in the virtual time of a simulation run before statistics gathering begins.

- ▶ Method of selecting an output path: **Base on probabilities**
 - Select a method that the process simulator will use to determine which processing path to follow when a process or an activity in a process has more than one set of outputs defined by output criteria.
- ▶ Recurring time interval for bundle creation: **2 Minutes** (270 in 9 hours)

Populating the simulation environment

Simulation attributes allow you to configure a process so that it behaves in a manner that resembles a real world business process.

A simulation environment is divided into four layers:

- ▶ The global simulation preferences (select *Windows@* → *Preferences* → *Business modeling* → *Simulation*) hold the default values for the local preferences of any newly created simulation snapshot.
- ▶ The local simulation (process default element) preferences are applied as default values for the simulation attributes of any new simulation profiles that you create for the current snapshot.
- ▶ The top-level process simulation attributes (process snapshot element) where you define the behavior of a process as a whole during a simulation.
- ▶ The low-level activity simulation attributes (process element or process snapshot element) where you define the behavior of an activity in a simulation.

Note: For more information about simulation level and information, refer to the product documentation under *Simulating Processes*.

With all information acquired, the first task is to assign a timetable and cost elements to the human role resources.

Note: The imported the model is already populated with the simulation information. You can go through the pages that follow to verify that the information is there.

Populating role resource information in the project

Roles add additional characteristics to resources. For example, an Employee resource could have the role of Customer Service Representative, Order Manager, or Shipper.

You can specify the roles that are required to complete any task that you model. You can add costs and availability to roles. For example, a certain role may cost \$20 an hour and be available only from Monday to Friday.

Customer service representative

In the Project Tree (Figure 6-5), navigate to the *Resources* and open the role that you want to modify (you have to insert data for the three resources).

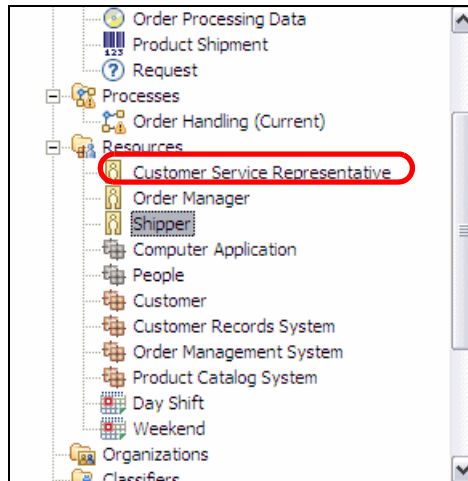


Figure 6-5 Open a resources role

In the Customer Service Representative editor area (Figure 6-6), insert the cost following this sequence of action:

- ▶ Select the *Costs* tab and click *Add*.
- ▶ Select *Cost per time unit* and click *OK*. The *Cost per unit time* appears in the list.
- ▶ Enter the cost value (11.00 USD for the Customer Service Representative).
- ▶ The time unit defaults to *1 hour*, which is what we need.

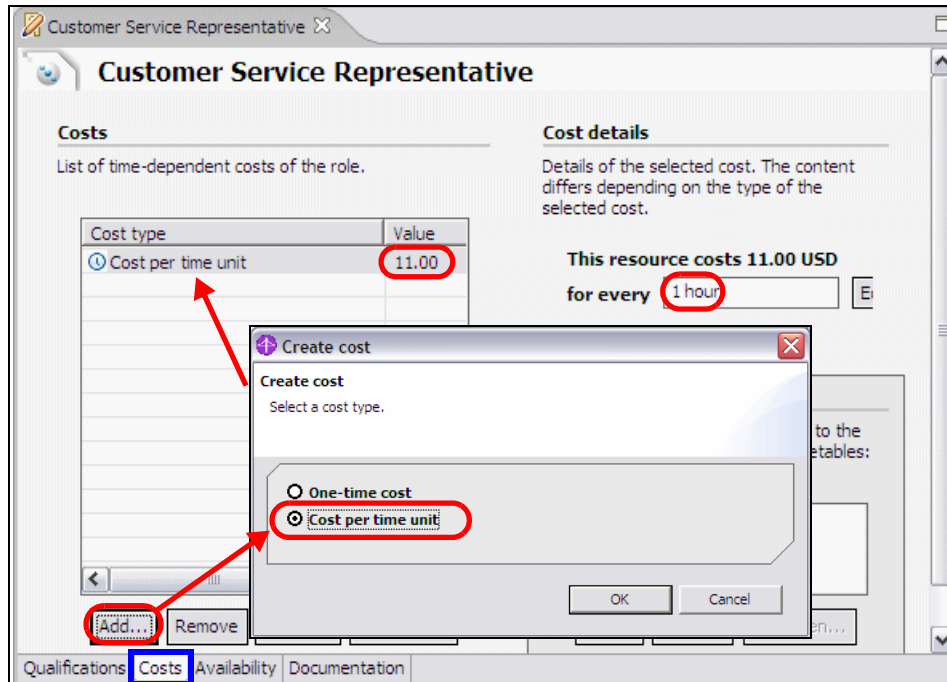


Figure 6-6 Role resource costs

In the Customer Service Representative editor, insert the duration (Figure 6-7):

- ▶ Select the *Availability* tab, and click *Add*.
- ▶ Select *Day Shift*, and click *OK*.
- ▶ Save the changes.

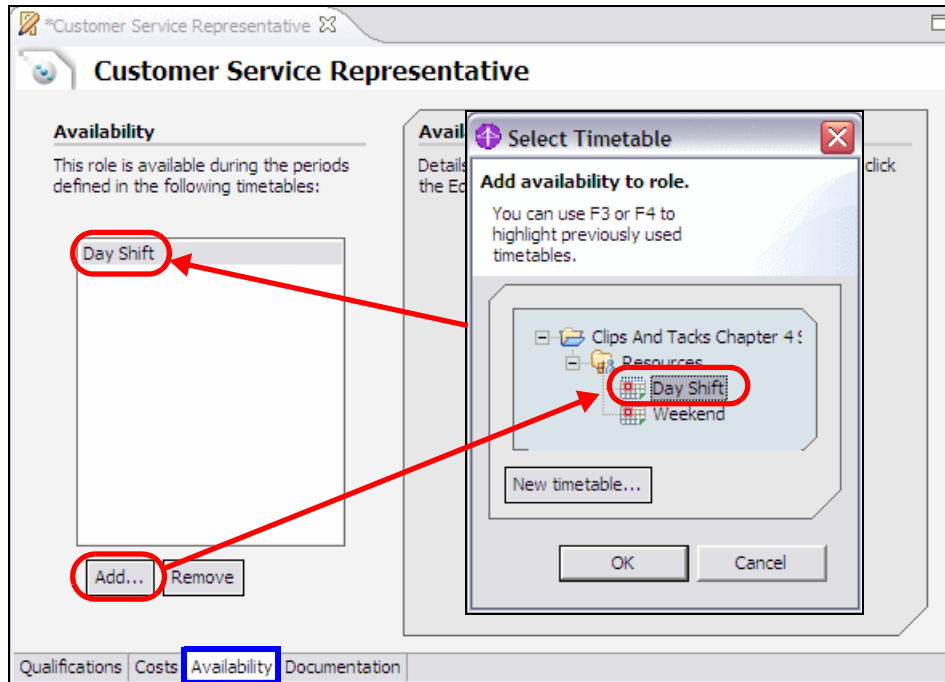


Figure 6-7 Role resource availability

Order Manager and Shipper

Perform the same operation for the Order Manager and Shipper roles and set the cost to \$20 and \$10, and the availability to Day Shift.

Populating duration information in the process

You can specify the processing time for an element such as a task or process. This is the time while the element is actively executing, rather than the elapsed time, which may include delays while waiting for a resource. For a task, you can also specify the maximum amount of time that the task should wait for a resource before failing.

In our case we will assign two types of duration information, one for the activity and one for the human tasks (role resource).

To be able to select individual activities, open the process diagram for the Order Handling (Current) process.

In the process flow select the activity, and in the Attributes view, select the *Duration* tab and enter the duration value (Figure 6-8).

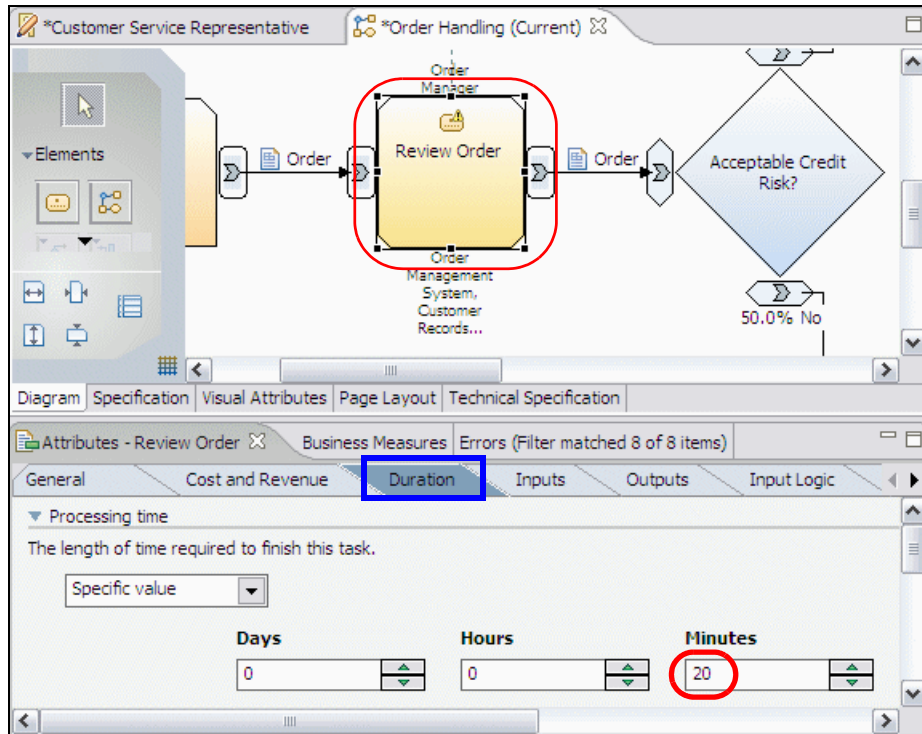


Figure 6-8 Activity duration

To populate the resource (human task) duration time (Figure 6-9):

- ▶ Select the activity in the model, for example, *Review Order*.
- ▶ Select the *Resources* tab in the Attributes view.
- ▶ Click Add to add a role.
- ▶ Click **...** in the *Role* column to select the role, for example, *Order Manager*.
- ▶ Click **...** in the *Time required* column and select the duration time, for example, 20 minutes.
- ▶ Save the changes.

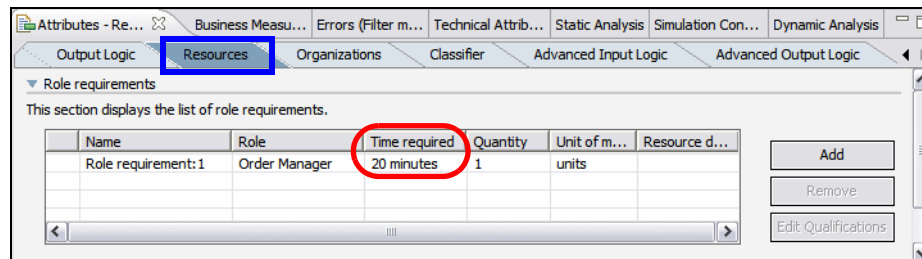


Figure 6-9 Activity role duration

The activity duration can have an overhead time on top of the human role duration time (this is the case for the Enter Account Number activity).

Important: In this example, you define duration times in the original flow, which is the master for the simulation snapshots. It is possible to change those values for each simulation without changing data in the original flow. This enables you to run multiple simulations without alteration of the original values that reflect the real world business process.

Validating the simulation data

You can validate in WebSphere Business Modeler if you have populated all the required information. Select the project and *Static Analysis* → *General Analysis* → *Matrix Analysis* (context). This function displays the roles by activities (Figure 6-10):

- ▶ Select *Activity* for rows, *Role* for columns, and click *Next*.
- ▶ Select *Processes*, select the process you want to analyze, and click *Finish*.

The screenshot shows the Matrix Analysis wizard with three panels. The first panel shows 'Row: Activity' and 'Column: Role'. The second panel shows 'Process catalogs' with 'Processes' selected. The third panel shows 'Process activity' set to 'Order Handling (Current)'. Below the wizard is a table showing the results of the static analysis.

	Order Manager	Shipper	Customer Service Representative
Order Handling (Current)			
Review Order	1 unit		
Cancel Order and Send Notification	1 unit		
Ship Order to Customer		1 unit	
Receive Order			
Determine if Customer has Existing Account			1 unit
Enter Customer Information and Assign Account N...			1 unit
Enter Account Number			1 unit
Enter Order Information			1 unit
Existing Account?			
Merge			
Acceptable Credit Risk?			

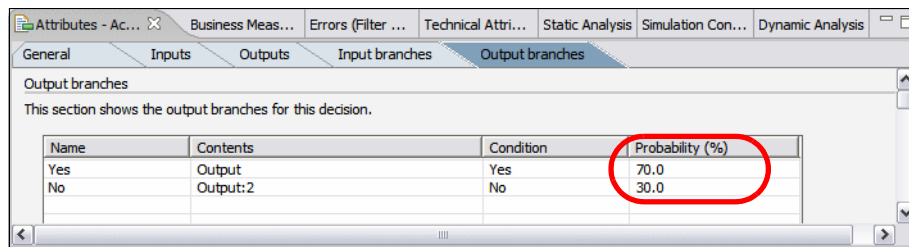
Figure 6-10 Static analysis: Activity versus role

Tip: After you create the simulation snapshot, you will be able to validate your process elements including durations, probabilities, and costs. See “Verify the profile specification” on page 108 for more information.

Populating probability information in the decision nodes

You can add a probability to each branch of a decision to indicate the probability of that branch.

The probability for each branch is set to 50% when the first two branches are created. You change original probabilities in the process to reflect the real world business process. In the process flow, select the Acceptable Credit Risk activity in the process flow. In the Attributes view, *Output branches* tab, set the probability values for the Yes and No branches (Figure 6-11). Note that you can also overtype the values in the process diagram.



Name	Contents	Condition	Probability (%)
Yes	Output	Yes	70.0
No	Output:2	No	30.0

Figure 6-11 Decision activity probability

Populating probability information in activity nodes

If you have more than one output criterion, you can optionally add probabilities to indicate how often each of the criteria occurs. For example, you could specify 60% for the more likely output and 40% for the less likely output. If you specify a probability for one output criterion, you must specify probabilities for all. If you do not specify probabilities, each output criterion is considered equally likely when you create a simulation snapshot.

For example, select the Ship Order to Customer activity in the process flow. In the Attributes view, Output Logic tab, you can see the three outputs as equal (Figure 6-12). To assign probabilities you would have to add rows and enter a probability for each output.

In our model we do not assign probabilities to the output of activity nodes.

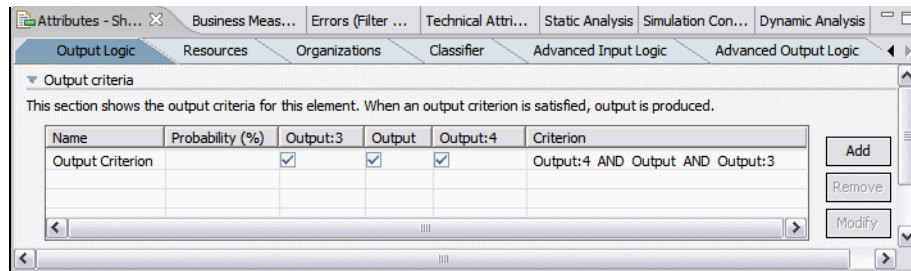


Figure 6-12 Activity output logic probability

Once you have populated every real world business process value in the simulation environment, you are ready to create a simulation snapshot. A snapshot reflects the real world simulation, including your assumptions.

Simulating the current process

In this section we simulate the current process and analyze the results.

Process instance simulation

Before running the simulation on your process, it is necessary to build a simulation snapshot and add information to get an accurate simulation.

The simulation information includes:

- ▶ The number of tokens for the simulation: **270 (one day)**
- ▶ The maximum duration of the benchmark: **365 days**
- ▶ The start time of the simulation: **Saturday, March 10, 2006 10:51:41 AM**
- ▶ The distribution model: **uniform distribution by minutes**
- ▶ The random number seed: **1**
- ▶ The steady delay for the process: **0 minute**
- ▶ The method of selecting an output path: **Based on probabilities to a single path**
- ▶ The recurring time interval for token creation: **2 Minutes (270 a day)**

Creating a simulation snapshot

When you simulate a process, the tool adds a simulation snapshot as a child element of the process in the project tree. A simulation snapshot is a record of the complete process model at the moment when you simulated the process. This record contains a copy of all the elements of your project that the process may use, such as business items, resources, and global tasks. You may want to

create multiple simulation snapshots for the same process after making changes to the project or to the process itself, so that you can compare the effect of these changes.

To create a simulation snapshot, select the Order handling (Current) process and *Simulate*.

When prompted if you want to validate the model, click *Yes* (Figure 6-13).

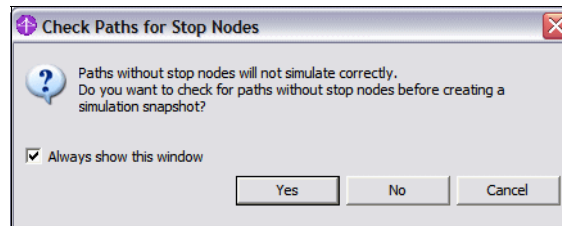


Figure 6-13 Check for paths without stop nodes

You will receive a warning that the validation cannot identify the process cases for this model (Figure 6-14). This is because the model includes an element of type repository for the Customer database, which prevents WebSphere Business Modeler from determining the finite set of process cases. For more details, see the article in the Info Center located at:

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.bt001s.help.modeler602.doc/doc/tasks/simulating/reviewforproccasesum.html>

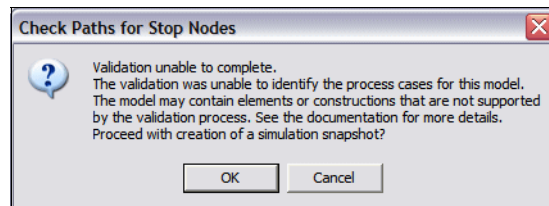


Figure 6-14 Warning from validation of the model

Within the simulation snapshot, the tool also creates two folders (Figure 6-15):

- ▶ **Simulation Snapshot™ Settings**—This is the folder with the name of the process plus “Simulation Snapshot Settings”, for instance, Order Handling (Current) Simulation Snapshot Settings. The simulation snapshot settings folder contains a set of local preferences for simulation attributes. When you create a new simulation profile for a simulation snapshot, the values specified in the local preferences are used for the simulation attributes of the process and activities within the process. The initial values of the local preferences are inherited from the global simulation preferences (*Windows* → *Preferences* → *Business Modeling* → *Simulation*).

- ▶ **Profile**—This is the folder with a timestamp appended to the process name, for instance, Order Handling (Current) Wednesday, February 28, 2007 9:31:46 AM PST. Each simulation snapshot contains an initial simulation profile. The simulation profile contains a copy of the process model at the time that you created the simulation snapshot. You can customize the process contained in this simulation profile, and you can create additional simulation profiles within the same simulation snapshot. Typically, you would create multiple simulation profiles for a simulation snapshot when you are experimenting with changes to the fields in the simulation profile, to determine the effect on process results.

The simulation profile opens as a process model named *Order Handling (Current) (Simulate)*. It looks similar to the process model and has queues added in front of the activities.

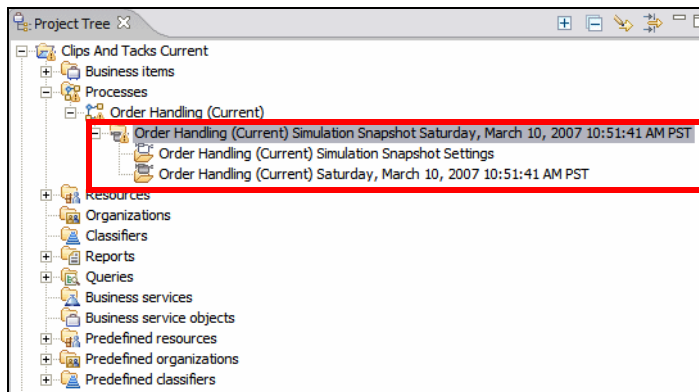


Figure 6-15 Warning from validation of the model

After you create the snapshot, you have to populate the system with your simulation data.

Simulation Snapshot Settings

Open the Order Handling (Current) Simulation Snapshot Settings and specify the values for your simulation as shown in Figure 6-16:

- ▶ In the *General* tab, verify the default values.
- ▶ In the *Token Creation* tab, select:
 - Recurring time interval for bundle creation: **2 minutes**

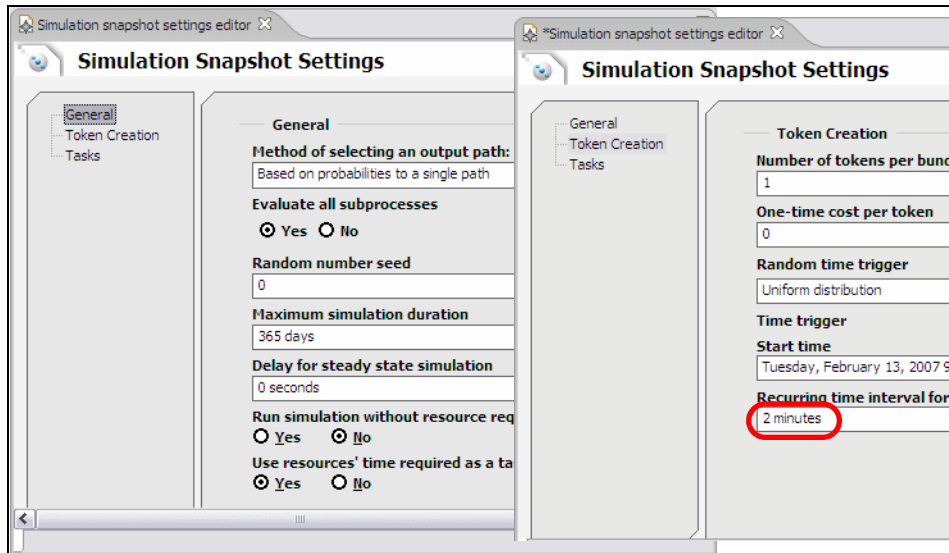


Figure 6-16 Process simulation default profile

Simulation snapshot process profile

The simulation profile should be open (after creation), but you can also open it from the Project Tree.

Important: To run and display a simulation properly, you have to set your computer time zone before a simulation. Your time zone should be equal to the time zone used in the project. In the current project, every element's time zone (timetable, simulation attributes) is defined as (GMT -8) Pacific Time. If you change your computer time zone, you have to restart the WebSphere Business Modeler.

- ▶ Populate the *General* tab (Figure 6-17):
 - Starting date (GMT-8), and ending date (GMT-8)
 - Evaluate all subprocesses: *Yes*
 - Maximum simulation duration: **365 days**
 - Random number seed: **1**
 - Delay of steady state simulation: **0 second**
 - Method of selecting an output path: *Based on probabilities to a single path*
 - Use resources' time required as a task processing time: *Yes*

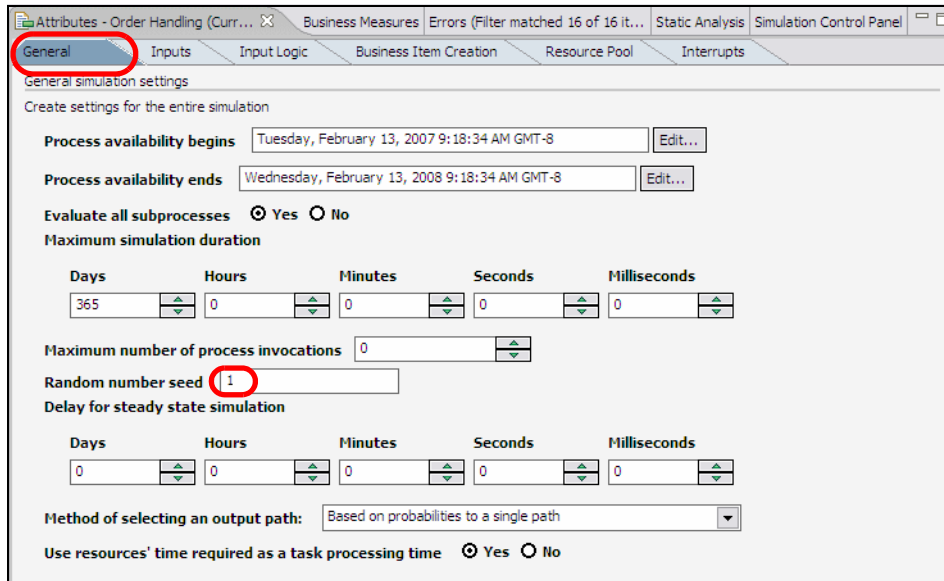


Figure 6-17 Simulation profile: General

- Populate the *Inputs* tab with the Total number of tokens, and Recurring time interval for bundle creation (Figure 6-18).

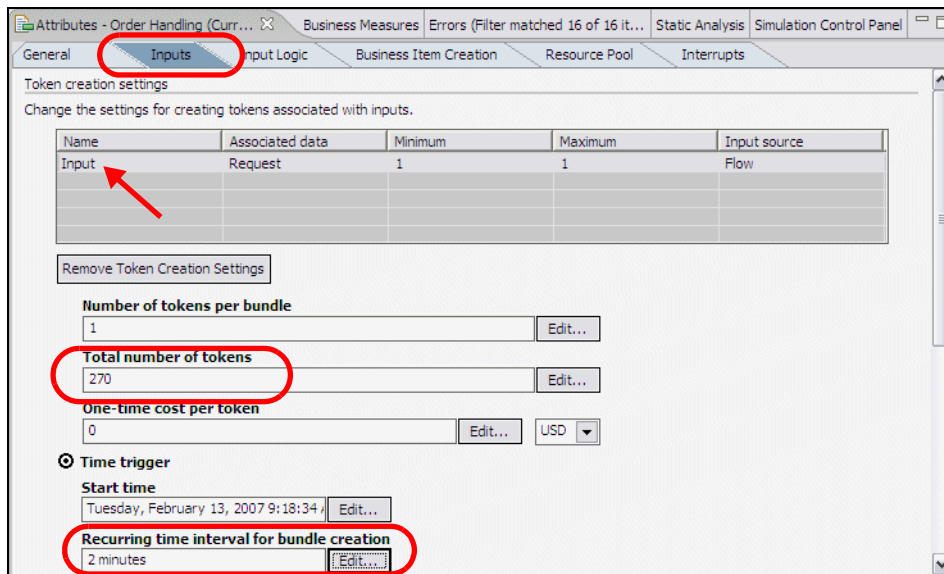


Figure 6-18 Simulation profile: Inputs

- Total number of tokens: **270** (you have to click *Edit*)
 - Select the *Time trigger*
 - Start time: same as the process starting date and time zone
 - Recurring time interval: **2 minutes**
- Finally, populate the number of human resources available in the *Resource pool* tab (Figure 6-19):
- For each role resource (Customer Service Representative, Order Manager, Shipper), deselect *Unlimited* and enter 1 in the quantity box.

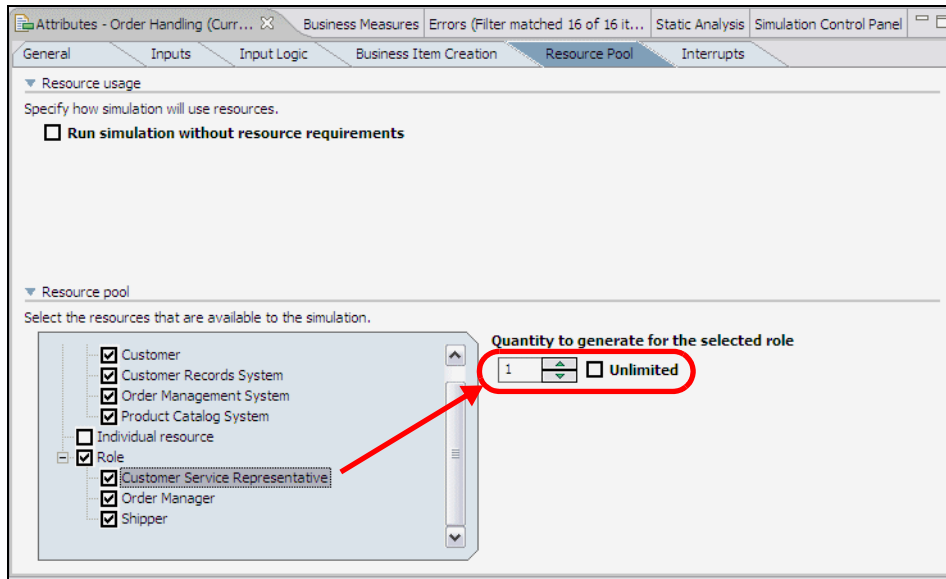


Figure 6-19 Simulation profile: Resource pool

Verify the profile specification

After saving the profile, you can verify the profile specification by selecting the Order Handling (Current) Profile and *Profile Analysis* → *Profile Specification*. Click *Select all* when prompted, and click *OK*.


The profile specification (Figure 6-20) lists all the processes with the resources that are used and the duration of each resource.

Note: You can also see the simulation attributes in the *Simulation Attributes* tab of the process (behind the *Diagram* tab).

Activity Name	Activity ...	Requiremen...	Resource or Role Name	Require...	Q...	Qu...	Inp...	Assoc...	Output...	Associate...
Acceptab...										
Cancel O...	2 minutes	Role Bulk Resource	Order Manager Order Management ...	2 minutes 2 minutes	1 1	unit unit	Input	Order	Output Output:2	Notification Order
Receive ...							Input	Request	Output:2	Order
Receive ...	20 seco...	Role Bulk Resource	Customer Service Re... Customer Records S...	20 seco... 10 seco...	1 1	unit unit	Input	Request	Output	Request
Receive ...	20 seco...	Role Bulk Resource	Customer Service Re... Customer Records S...	10 seco... 5 seconds	1 1	unit unit	Input	Request	Output	Request
Receive ...	6 minutes	Role Bulk Resource	Customer Service Re... Customer Records S...	5 minut... 15 seco...	1 1	unit unit	Input	Request	Output:2 Output	Customer ... Request
Receive ...	1 second	Role Bulk Resource	Customer Service Re... Product Catalog Sys...	12 minu... 0 seconds	1 1	unit unit	Input	Request	Output	

Figure 6-20 Profile analysis: Processes with resources and duration

Running the simulation

We are now ready to run the simulation. Open the *Simulation Control Panel* view (Figure 6-21) behind the *Attributes* area, and click the green arrow icon  to start the simulation.

Simulation ready to run, based on saved simulation settings.




Time that simulation has been running: 00:00:00

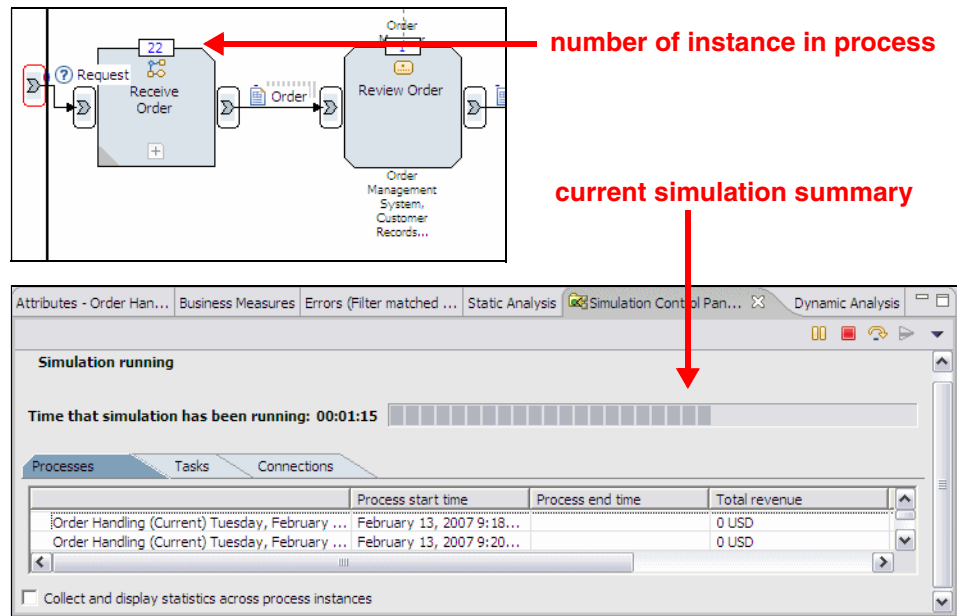
Process start time	Process end time	Total revenue	Tot

Collect and display statistics across process instances

Figure 6-21 Simulation control panel

Note: If you see a message that the simulation is unavailable you have to close the editor with the simulation snapshot settings.

The system shows the token moving in the flow and the list of process instances (Figure 6-22). At this point, you might pause , stop , or step through  the simulation.



number of instance in process

current simulation summary

Processes	Tasks	Connections
Order Handling (Current) Tuesday, February ...	February 13, 2007 9:18...	0 USD
Order Handling (Current) Tuesday, February ...	February 13, 2007 9:20...	0 USD

Figure 6-22 Current process simulation running

Tip: You can run the simulation faster without animating the simulation (Figure 6-23).

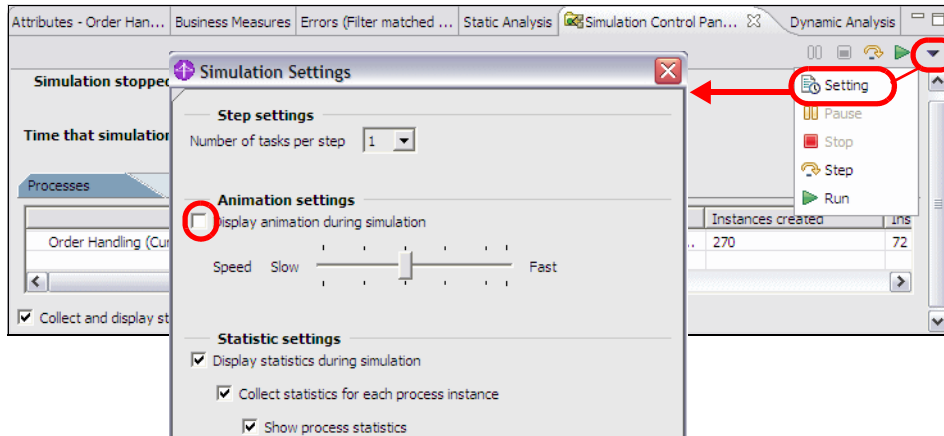


Figure 6-23 Simulation settings

Analyzing the simulation results (current)

A first glance in the Simulation Control Panel Tasks tab (Figure 6-24) shows that 207 of the 270 orders went through shipping, and 63 were cancelled. Therefore the shipping percentage is 76.7%.

	Total instances	Active instances	Average revenue	Average cost
Cancel Order and Send Notification	63	0	0 USD	0.667 USD
Determine if Customer has Existing Account	270	0	0 USD	0.061 USD
Enter Account Number	270	0	0 USD	0.031 USD
Enter Customer Information and Assign Acco...	139	0	0 USD	1.054 USD
Enter Order Information	270	0	0 USD	2.2 USD
Review Order	270	0	0 USD	6.667 USD
Ship Order to Customer	207	0	0 USD	2.5 USD

Figure 6-24 Simulation result tasks

Once the simulation is complete, you can now use the dynamic analysis function on the simulation result element (Figure 6-25).

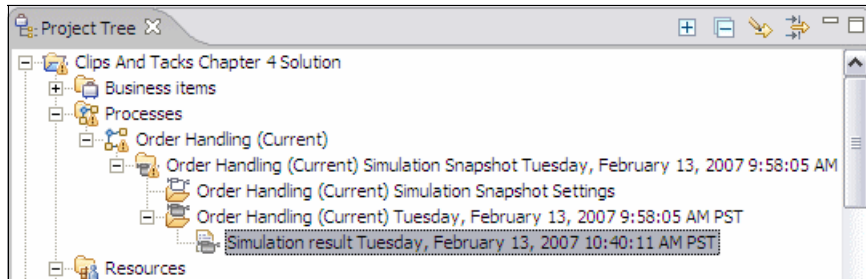


Figure 6-25 Simulation result element

For the current case, the business analyst needs four categories of the process:

- ▶ Process duration
- ▶ Process cases summary
- ▶ Resource usage
- ▶ Process cost

Process duration

To get the process duration information, select the simulation result element and *Dynamic Analysis* → *Process Cases Analysis* → *Process Duration*:

- ▶ When prompted, select *All process instances*.
- ▶ When prompted, click *Yes* for the Path Signatures.

This analysis shows process elapsed duration and throughput details for each process case in a simulation.

Process elapsed duration is the duration that a process case takes if started at a specific time and date. Process elapsed duration includes transfer times between activities and the elapsed durations of all activities on the critical path. The critical path is defined as the processing path that has the longest duration of all parallel paths in the process case. Calculations are performed per case by getting the simple average of the process instances duration records in a case.

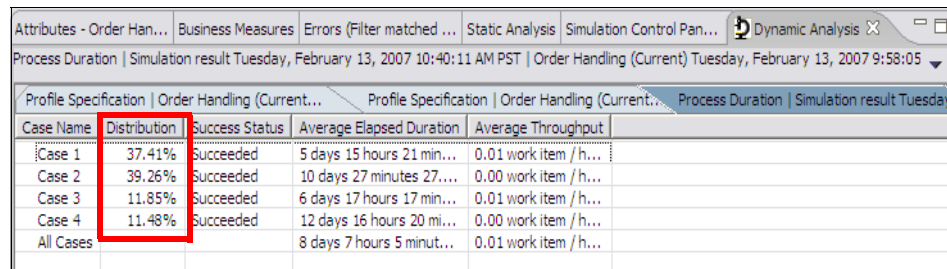
You can use this analysis when you want to examine process level processing durations and throughputs for each generated process case. This analysis, like other process case analyses, may reveal unexpected results within specific process cases.

For example, you may determine that the average throughput in a particular process case is unacceptably low. As a result of reviewing the information that this analysis presents, you may decide that you need to modify the process model or reset resource levels, or you may determine that you want to investigate

further with another type of process case analysis such as process resource analysis.

Alternatively, you can examine the duration results for specific process instances within a process case by running the process instance summary analysis and then the process instance time analysis.

In the summary, the process instances analysis shows four cases that reflect four different ways of processing of customers requests (Figure 6-26).



Case Name	Distribution	Success Status	Average Elapsed Duration	Average Throughput
Case 1	37.41%	Succeeded	5 days 15 hours 21 min...	0.01 work item / h...
Case 2	39.26%	Succeeded	10 days 27 minutes 27....	0.00 work item / h...
Case 3	11.85%	Succeeded	6 days 17 hours 17 min...	0.01 work item / h...
Case 4	11.48%	Succeeded	12 days 16 hours 20 mi...	0.00 work item / h...
All Cases			8 days 7 hours 5 minut...	0.01 work item / h...

Figure 6-26 Process duration analysis

Note: Your results will be different than ours. Every simulation produces results that vary.

Select a case and a blue line in the process diagram shows the path:

- ▶ The first case reflects a **shipped** product for a new customer (with new customer entry in the company database).
- ▶ The second case reflects a **shipped** product for an existing customer.
- ▶ The third case reflects a **cancelled** order for a new customer
- ▶ The fourth case reflects a **cancelled** order for an existing customer.

At this point, we can validate that of 270 requests, only 76.7% (37.41 + 39.26) were shipped to customers.

One of the goals of the improvement will be to increase this percentage.

Another type of analysis can be done with the simulation tool, if you know how long your customers can wait for a product and you want to know how long your company can wait before taking corrective action. You can run the simulation for many durations (1 day, 2 days, 1 week, 2 weeks, 4 weeks, 8 weeks). As a result you can see the curve of order handling duration and the average delay to ship a product to a customer (Figure 6-27).

For instance, if your customer can wait 100 days for a product, we have to take an action (for example, add a new shipper) after 4 weeks of running. Or we can choose to improve your process to get a more linear result.

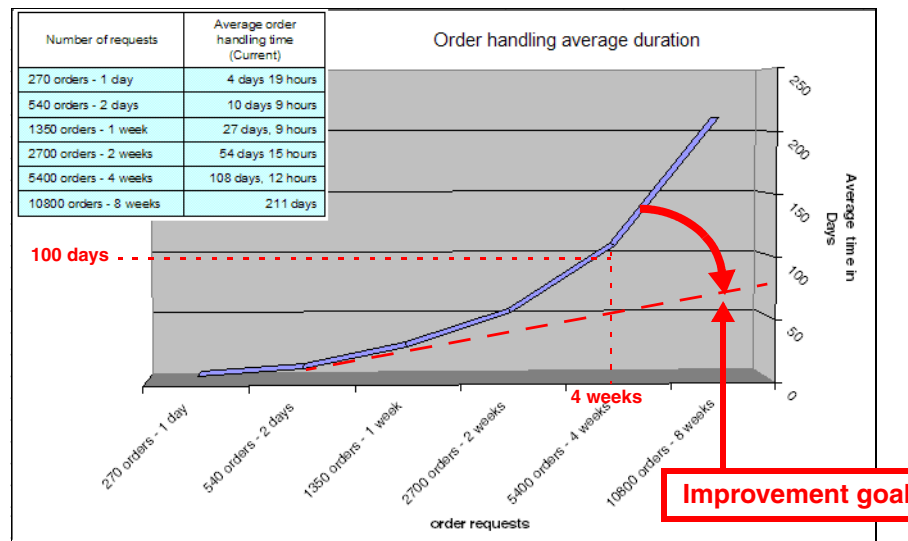


Figure 6-27 Order handling average duration curve

Conclusion of the business analyst

The business analyst detects an abnormal behavior and a bottleneck in these results. He should analyze the worst case to figure out the resource problem. Now we know that we cannot sustain the current process in the long run.

Process cases summary

To see the process cases summary, select the simulation result element and *Dynamic Analysis* → *Process Cases Analysis* → *Process Cases Summary*.

The process cases summary analysis (Figure 6-28) shows summary details for all the process cases produced during the simulation of a process. A process case is defined as a set of process instances that have the same processing path.

You can use this analysis when you want an overview of the process cases generated in a simulation. This analysis provides high level summary information for each process case, including duration and cost information and an indicator whether the process case was successful or not. This analysis lists the activities completed in each case and quantifies their average total cost and average total elapsed duration.

This analysis—like other process case analysis—might reveal unexpected results within specific process cases. For example, you might determine that the average process elapsed duration in a particular process case is unacceptably high. As a result of reviewing the information that this analysis presents, you could decide that you need to modify a process model or reset resource levels, or you might determine that you want to investigate further with another type of process case analysis or process instance analysis. Alternatively, you can investigate the reasons that cause a particular process case to fail.

Case N...	Average Cost	Average Elapsed Duration	Average Working ...	Total Cost	Total Reso...	N...	Distribution	Success Status
Case 1	USD 11.46	5 days 15 hours 21 min...	47 minutes 30 sec...	USD 1,157.29	USD 1,157.29	101	37.41%	Succeeded
Case 2	USD 12.51	10 days 27 minutes 27....	53 minutes 15 sec...	USD 1,326.33	USD 1,326.33	106	39.26%	Succeeded
Case 3	USD 9.62	6 days 17 hours 17 min...	34 minutes 30 sec...	USD 308.00	USD 308.00	32	11.85%	Succeeded
Case 4	USD 10.68	12 days 16 hours 20 min...	40 minutes 15 sec...	USD 331.05	USD 331.05	31	11.48%	Succeeded
All Ca...	USD 11.57	8 days 7 hours 5 minut...	47 minutes 23.055...	USD 3,122.67	USD 3,122.67	270	100.00%	

Figure 6-28 Process cases summary

If we analyze details of the two worst cases (case 2 and case 4), we can see what the time consuming activities are (Figure 6-29).

Case Name	Activity Name	Average Cost	Average Elapsed Duration	Average
Case 1		USD 11.46	5 days 20 hours 29 minutes 21.864 seconds	47 m
Case 2		USD 12.51	10 days 18 hours 36 minutes 59.778 seconds	53 m
	Order Handling (Current)	USD 12.51	10 days 18 hours 36 minutes 59.778 seconds	53 m
	Acceptable Credit Risk?	USD 0.00	0 seconds	
	Receive Order	USD 3.35	7 days 7 hours 26 minutes 9.201 seconds	18 m
	Determine if Customer has Existing Account	USD 0.06	13 hours 2 minutes 57.442 seconds	
	Enter Account Number	USD 0.03	3 days 3 hours 40 minutes 0.634 seconds	
	Enter Customer Information and Assign Account Number	USD 1.05	1 day 2 hours 25 minutes 17.75 seconds	5 m
	Enter Order Information	USD 2.20	2 days 12 hours 17 minutes 53.375 seconds	
	Existing Account?	USD 0.00	0 seconds	
	Merge	USD 0.00	0 seconds	
	Review Order	USD 6.67	3 days 9 hours 45 minutes 18.461 seconds	
	Ship Order to Customer	USD 2.50	1 hour 25 minutes 32.115 seconds	
Case 3		USD 9.62	7 days 4 hours 43 minutes 47.156 seconds	34 m
Case 4		USD 10.68	12 days 22 hours 11 minutes 14.516 seconds	40 m
	Order Handling (Current)	USD 10.68	12 days 22 hours 11 minutes 14.516 seconds	40 m
	Acceptable Credit Risk?	USD 0.00	0 seconds	
	Cancel Order and Send Notification	USD 0.67	1 day 15 hours 18 minutes	
	Receive Order	USD 3.35	7 days 18 hours 55 minutes 58.387 seconds	18 m
	Determine if Customer has Existing Account	USD 0.06	13 hours 43 minutes 14.322 seconds	
	Enter Account Number	USD 0.03	3 days 5 hours 26 minutes 39.193 seconds	
	Enter Customer Information and Assign Account Number	USD 1.05	1 day 5 hours 16 minutes 10.032 seconds	5 m
	Enter Order Information	USD 2.20	2 days 18 hours 29 minutes 54.838 seconds	
	Existing Account?	USD 0.00	0 seconds	
	Merge	USD 0.00	0 seconds	
	Review Order	USD 6.67	3 days 11 hours 57 minutes 16.129 seconds	
All Cases		USD 11.56	8 days 17 hours 17 minutes 42.622 seconds	47 m

Figure 6-29 Process cases 1 and 4 analysis in detail

We find that the high consuming activities are Enter Account Number and Enter Order Information. The Receive Order subprocess, which contains the two consuming activities, is the first bottleneck of the current process.

Conclusion of the business analyst

Several customer surveys have indicated that ClipsAndTacks' customers are not satisfied with the ordering process. The telephone ordering procedure is time-consuming, and customers are frustrated at being placed on hold while waiting for the next available representative. Regular customers are frustrated at the amount of time it takes to receive their orders. These delays are caused by the enter order information process, requiring a lot of time from the customer service representative. The process cases summary enables us to assign the delays to the enter order information and account number activities. Now we have to look inside those activities to figure out which resource is an issue.

Resource usage

To see the resource usage, select the simulation result element and *Dynamic Analysis* → *Aggregated Analysis* → *Resource Usage*.

This analysis shows information about usage of each resource that is allocated in a process simulation.

This analysis helps in resource planning, as it enables you to see how each resource is allocated to different activities across the process. In addition to showing how a resource uses its time to accomplish one or more activities, this analysis shows where shortages of resources cause delays in completing activities. You can use this information to determine where additional resources are required.

We take a look at the enter order information activity (Figure 6-30) to identify which resources to analyze. In this activity two resources are used. One resource is a bulk resource, so this resource cannot be the bottleneck. The other resource is the customer service representative. We have to look in the resource usage analysis to see if this resource has high shortage duration.

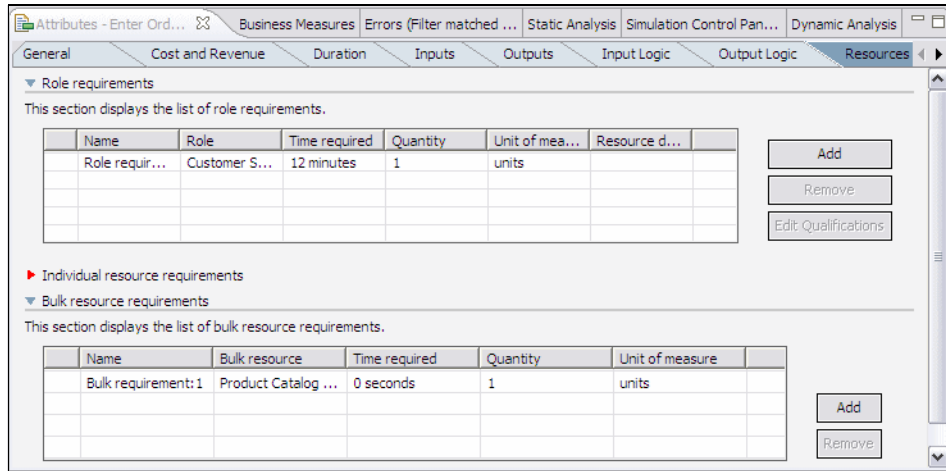


Figure 6-30 Handling Order (Current): Review Order: Enter order information

The resource usage sheet (Figure 6-31) shows the exponential growth of the shortage duration for the customer service representative. This demonstrates that the customer service representative is really the bottleneck of this process.

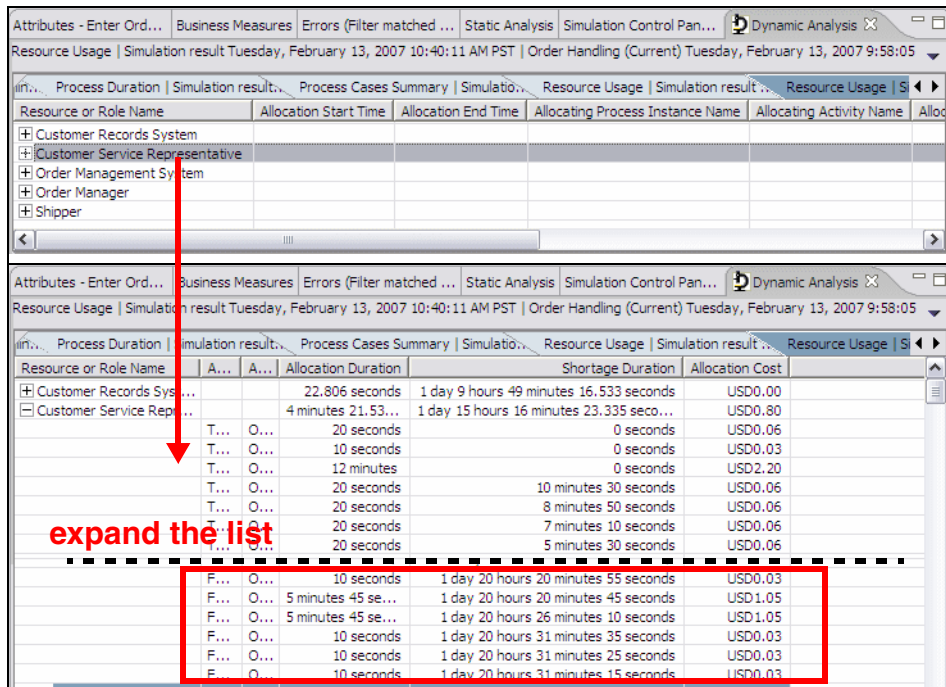


Figure 6-31 Customer service representative shortage duration

Conclusion of the business analyst

This analysis shows the overbooking of the customer service representative and the result in term of shortage durations. To fix this problem, two solutions can be proposed:

- ▶ Add a new customer service representative
- ▶ Transfer activities from the customer service representative to the customer by introducing a Web application to enter orders.

In response to this bottleneck, the new process will eliminate the need for contact between customers and customer service representatives when an order is placed. Customers will be able to browse the ClipsAndTacks product catalog and enter their own order information using a Web application. New customers will be able to enter their company information and receive a customer account number immediately.

Customers who have an account number will be able to enter it and prompt the Web application to retrieve their information and pre-fill the Web form with their address and preferred shipping information. The new Web application, including the product catalog and order form, will be available 24 hours a day, 7 days a week.

Process cost

To see the process cost, select the simulation result element and *Dynamic Analysis* → *Process Cases Analysis* → *Process Cost*.

The process cost analysis shows the average cost and revenue for all process instances in each case in the current simulation result, and the weighted average costs and revenues for all process cases (Figure 6-32).

You can use this analysis when you want to examine process level costs and revenues for each generated process case. This analysis, like other process case analyses, may reveal unexpected results within specific process cases.

For example, you might determine that the average profit in a particular process case is unacceptably low. As a result of reviewing the information that this analysis presents, you could decide that you need to modify a process model. You can also examine the cost results for specific process instances within a process case by running the process instance summary analysis and then the process instance cost analysis.

Case Name	Distribution	Success ...	Avera...	Avera...	Avera...	Average Resource Cost	Average Cost	Average Profit
Case 1	37.41%	Succeeded	USD0.00	USD0.00	USD0.00	USD11.46	USD11.46	(USD11.46)
Case 2	39.26%	Succeeded	USD0.00	USD0.00	USD0.00	USD12.51	USD12.51	(USD12.51)
Case 3	11.85%	Succeeded	USD0.00	USD0.00	USD0.00	USD9.62	USD9.62	(USD9.62)
Case 4	11.48%	Succeeded	USD0.00	USD0.00	USD0.00	USD10.68	USD10.68	(USD10.68)
All Cases			USD0.00	USD0.00	USD0.00	USD11.57	USD11.57	(USD11.57)


Figure 6-32 Process cost analysis

During this simulation the average costs were:

- ▶ \$11.46 to ship a product for a new customer
- ▶ \$12.51 to ship a product for an existing customer
- ▶ \$9.62 to cancel an order for an existing customer
- ▶ \$10.68 to cancel an order for a new customer
- ▶ The average cost for the process is \$11.57

Note: Suppose you want to know the number of orders cancelled by customers. You assume how long the customers can wait for a product. Then populate the duration limit in the Ship Order to Customer activity, *Duration* tab, *Resource wait time* field. With this parameter, every order not shipped due to a very long processing duration will be considered a failed instance.

Closing the reports

To close a report, use the arrow pull-down  in the Dynamic Analysis view and select *Close* or *Close All*.

Other reports

There are many more dynamic analysis reports that you can explore. Experiment with the Dynamic Analysis menus to view more reports.

More information

For more information about Analysis, refer to the product documentation under *Analyzing models and simulations*.

Summary

The simulation and analysis of the current process model shows significant bottlenecks that must be reduced to keep the current customers happy and to handle future increase in customer demand.

The model with simulation snapshot is exported as:

SG247148\sampcode\model\Clips And Tacks Current with Simulation.mar



Modeling the Future 1 business process

This chapter describes how the ClipsAndTacks current process is modified to become the Future 1 process.

The key to a successful transformation of the current process is to define step by step all the information acquired by the business analyst.

The first part of this chapter describes the result of the current process analysis.

The second part of this chapter shows how the business analyst modifies the process step-by-step to build the new process using WebSphere Business Modeler. No simulation information is populated at this stage.

The third part of this chapter shows how the IT architect populates the technical information in accordance to the BPEL technical constraints. Typically the business analyst would consult with the systems architect to discuss implementation issues.

Documenting the Future 1 process

The Future 1 model enables an organization to capture the potential results of any changes it makes to its process. The Future 1 model not only provides simulation data and analysis; it serves as a blueprint for the solution architect and programmers whose responsibility it is to create and implement the new runtime process.

Business revision

The management of ClipsAndTacks wants to improve the company's revenue by improving its order handling process. The assumption is that a shorter wait time for orders and fewer rejected orders will improve customer satisfaction and result in increased new and repeat business for the company.

In the ClipsAndTacks scenario, you will use the current process model as the starting point for the planned revisions to the process model. These are the key changes in the revised process:

- ▶ A Web application enables customers to enter their own account and order information.
- ▶ Orders are sent for review or shipping depending on a new business rule, where the threshold value for automatically approved orders is raised to \$750.

The business analyst and the management team have determined that the current order handling rule—that is, orders over a certain amount must be approved by an order manager—should remain in place in the new process, but that the rule has to be enhanced. The threshold will be raised from \$500 to \$750 to reduce the number of orders requiring approval. The business analyst believes that the raised threshold will reduce demand on the order approver, thus speeding up the approvals process for those orders that require it, and reducing the average order completion time.

Summary of revisions:

- ▶ In the Receive Order process, customer service representatives are replaced by a single Web Application that provides customer and product information (shorten order process).
- ▶ Orders are accepted 24 hours a day, 7 days a week.
- ▶ Customer completes the Web order form and submits the order; no customer service representatives are required in this process (reduce labour costs).
- ▶ Orders are checked automatically against a business rules engine.
- ▶ Regular customer orders over \$750 must be approved by the order manager (shorten average order time and increase percentage of approved orders).

Flow revision

The new order handling process contains many of the same activities, but the actors in the process have changed. The process proceeds as shown in Figure 7-1.

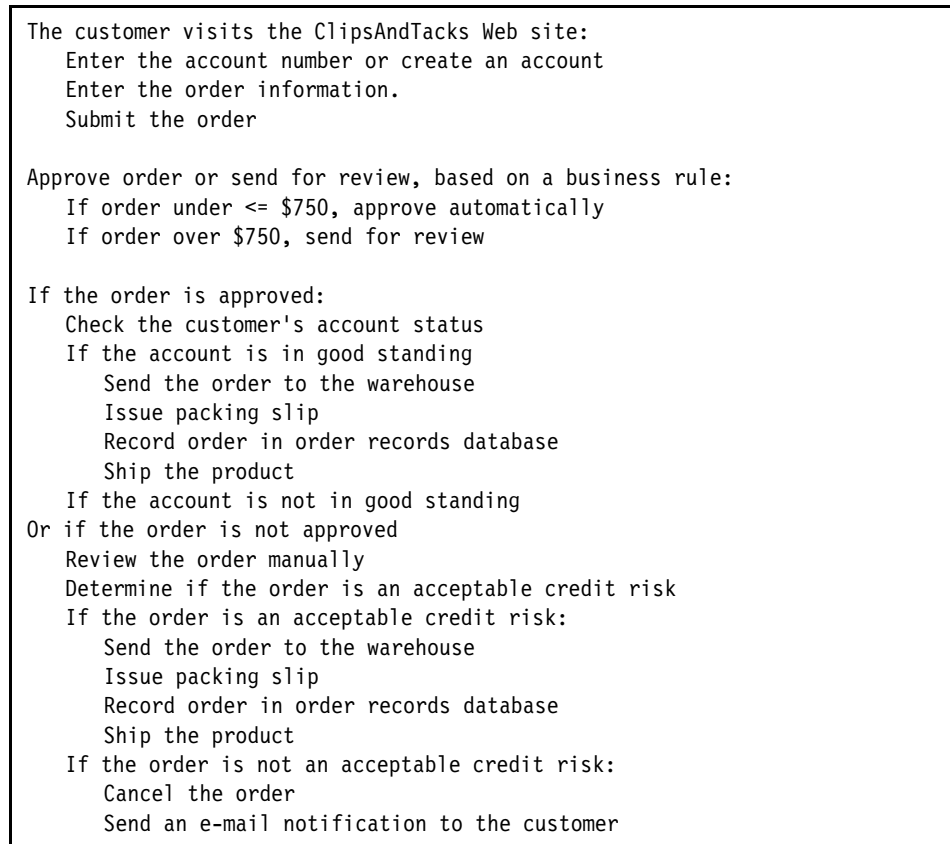


Figure 7-1 Order handling process description

Resource items revision

To model the revised process, the business analyst requires that the following model elements be created:

- ▶ Web application resource—Replaces customer service representatives, customer records system, and product catalog system in the Receive Order process.
- ▶ Online application timetable—The 24/7 timetable on which the new Web application operates.

Technical outline of the implementation in the Modeler

We make a copy of the current process model and rename it to **Order Handling (Future 1)**. We make the Receive Order process a global process and we update each task that previously required the customer service representative to access the product catalog system, the customer records system, and the order management system. We replace the representatives and various systems with a single Web application that will handle the product catalog, customer records, and the customer order. We also update the model to show that the customer is now a resource used in the process.

Building the Future 1 process

In this section, we explain how to build the new process step-by-step:

- ▶ Create new timetable resource
- ▶ Create new non-consumable bulk resource
- ▶ Create the new process Order Handling (Future 1)
- ▶ Create new activities
- ▶ Create new decisions
- ▶ Create merge nodes
- ▶ Create connections
- ▶ Populate resources in activities
- ▶ Populate expressions in decisions
- ▶ Populate business comments
- ▶ Validate the process
- ▶ Organize the diagram

Note: You can import the revised model from:

```
SG247148\sampcode\model\Clips And Tacks Future 1.mar
```

See “Importing the current process model using the Modeler” on page 83 for instructions on how to import a model.

To simplify the model, we removed the computer applications: Customer Records System, Order Management System, and Product Catalog System. They have little impact on simulation and no impact on implementation.

Creating timetables

A timetable is a schedule of times. In business process modeling, timetables are usually associated with resources or costs. A timetable for a resource indicates availability (such as Monday to Friday). A timetable for the cost of a resource indicates when the cost applies.

You can set up timetables containing recurring time intervals that are relevant to the business. For example, you may want to model costs of resources that vary depending on the time of day, such as electricity, or costs of resources that vary depending on the time of year, such as seasonal workers.

In this section, we create a timetable to define the online time of the future Web application and a new timetable to define the frequency of customer requests.

Creation of the Online Application timetable

In the Project Tree, select *Resources* and *New* → *Timetable*.

In the dialog box (Figure 7-2), enter the name *Online Application* and click *Finish*.

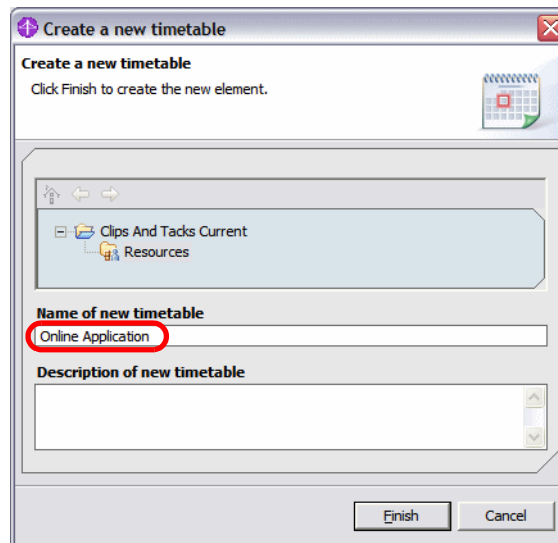


Figure 7-2 Create timetable dialog

Once you have created the new timetable, you have to populate all parameters (Figure 7-3):

- ▶ Leave *Forever* selected and leave the repetition period as 1 day.
- ▶ Click *Select Time*.
- ▶ In the *DateTime* dialog box, set the time zone (GMT-8), year (2006), month (July), day (18), hour (0), minutes (0), seconds (0) and A.M. The date and time specified here should be less than or equal to the starting the date of your future simulation. Click *OK*.

Note: The time zone must be equal to the time zone defined in the rest of the project and your computer time zone. The value 12:00:00 A.M stands for midnight.

- ▶ Click *Select duration*, enter 24 hours, and click *OK*.
- ▶ In the Selected interval details area, set the same time as above (July 18, 2006, 0am).
- ▶ Save and close the dialog.

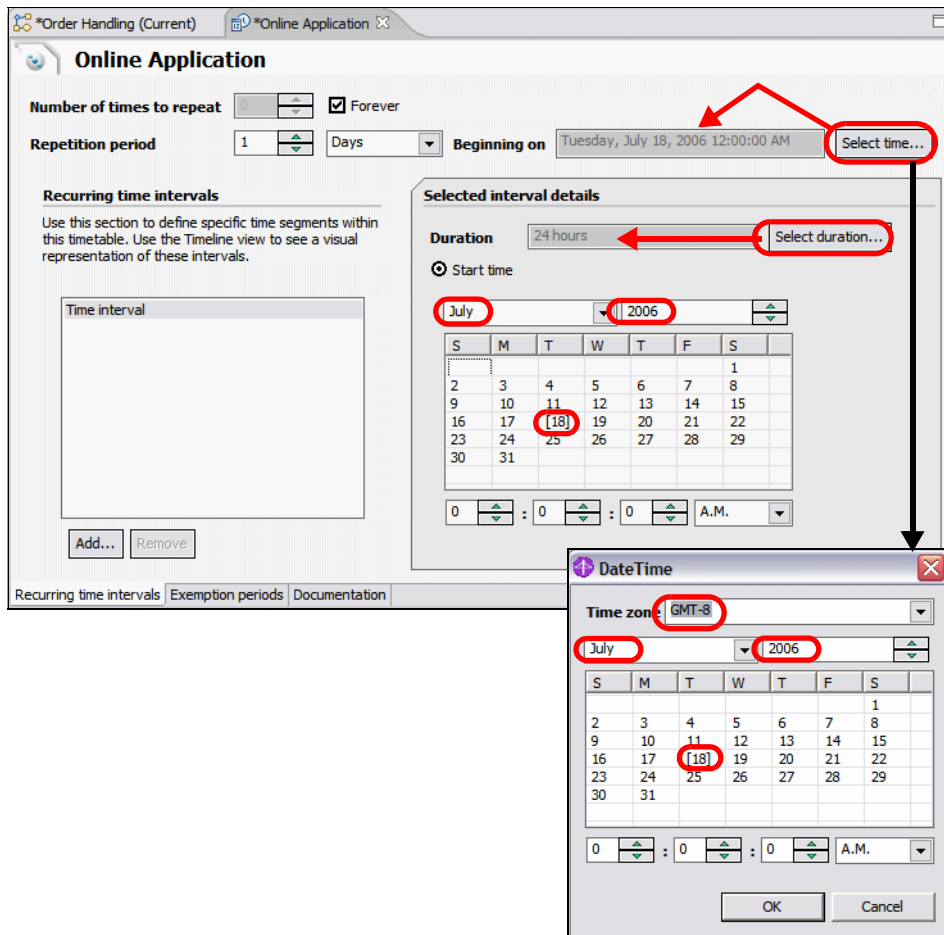


Figure 7-3 Online Application timetable

Creation of the Online Request timetable

Create another timetable in the same way and name it Online Request.

Populate the Online Request timetable with the same values as the Online Application timetable (Figure 7-3).

Add an exemption period

Select the *Exemption Periods* tab (Figure 7-4) and click *Add*. Select *Weekend* and click *OK*. The data of the Weekend is filled in the right-hand side.

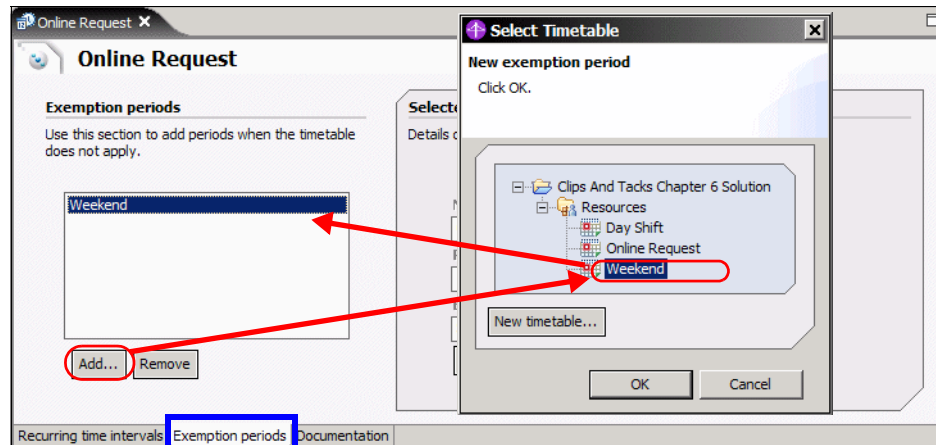


Figure 7-4 Online request timetable, exemption tab

Creating a new non-consumable bulk resource

Resources represent the people, equipment, or material used to perform a project or a task. Examples of resources are Computer and Employee.

You can model two types of resources: individual resources and bulk resources:

- ▶ **Individual resources** are resources where a specific instance is required, whereas bulk resources are resources where any instance of a type of resource from a pool can be used. Individual resources include people and computers, and bulk resources include power and water.
- ▶ **Bulk resources** can represent the material used to perform a project or a task. They can be non-consumable (such as employees, vehicles, or equipment) or consumable (such as fuel or printer paper). Consumable resources are diminished, or perhaps even used up, during the process.

You can define bulk resources as resources that are not uniquely identified, but whether resources need to be identifiable may depend on how they are being used. In a car rental agency, the cars can be viewed as either bulk or individual resources. From an executive's perspective, individual cars are not of interest, but bulk information is. At the rental desk, the individual resource information of each specific car is important. The modeling of a resource therefore depends on the process being modeled and its purpose.

Resources may have specified periods when they are available. To specify availability, you can use an existing timetable or create a new timetable that indicates the periods of availability. If you do not specify the availability of a resource, it is assumed that the resource is always available. You can also add costs and qualifications to resources. The qualifications are specific roles that this resource must fulfill.

In this section we create a new bulk resource that describe the operational working time and the number of user sessions for the Web application:

- ▶ Select *Resources* and select *New* → *Resource*.
- ▶ In the dialog (Figure 7-5), enter the name *Web Application*, select *Computer Application* for Associated resources definition, select *bulk* as Resource type, and click *Finish*.

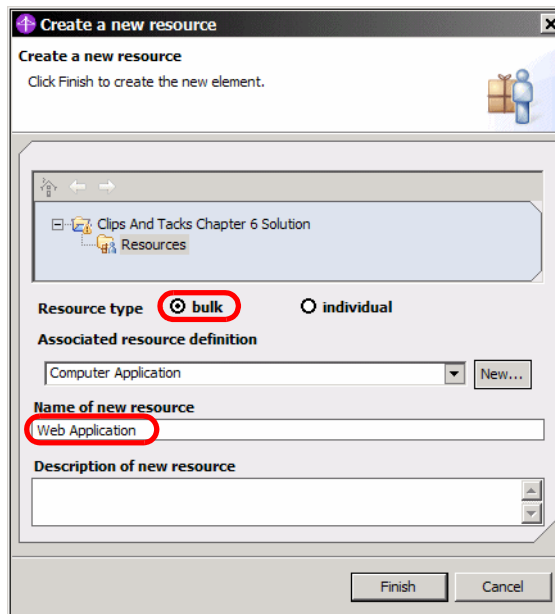


Figure 7-5 Create resource dialog

- ▶ Populate the Web Application resource with the cost (Figure 7-6):
 - Select the *Costs* tab.
 - Click *Add*.
 - Select *Cost per quantity and time unit* and click *OK*.
 - Enter the cost as 1 USD.
- ▶ Populate the Web Application resource with the availability:
 - Select the *Availability* tab.
 - Click *Add*.
 - Select *Online Application* and click *OK*.
 - Set the *Available quantity* (which in our case is the number of user sessions) to 1000.

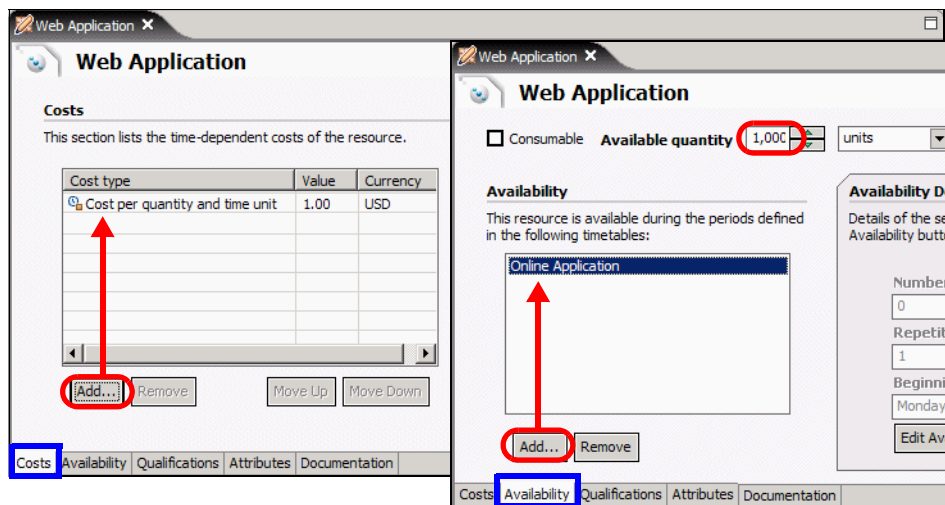


Figure 7-6 Web Application bulk resource cost and availability

Creating the new process: Order Handling (Future 1)

In this section we copy the current process to create a new process renamed to Order Handling (Future 1):

- ▶ Expand Processes, select the *Order Handling (Current)* process and *Copy*.
- ▶ Select *Processes* and *Paste*. Now you have a new process named *Copy of Order Handling (Current)*.
- ▶ Rename the process by selecting the copied process and *Rename*. Enter *Order Handling (Future 1)* as new name.

Overview of the new process

The finished process is shown in Figure 7-7 (see also Figure 7-22 on page 145). We will implement the process in stages.

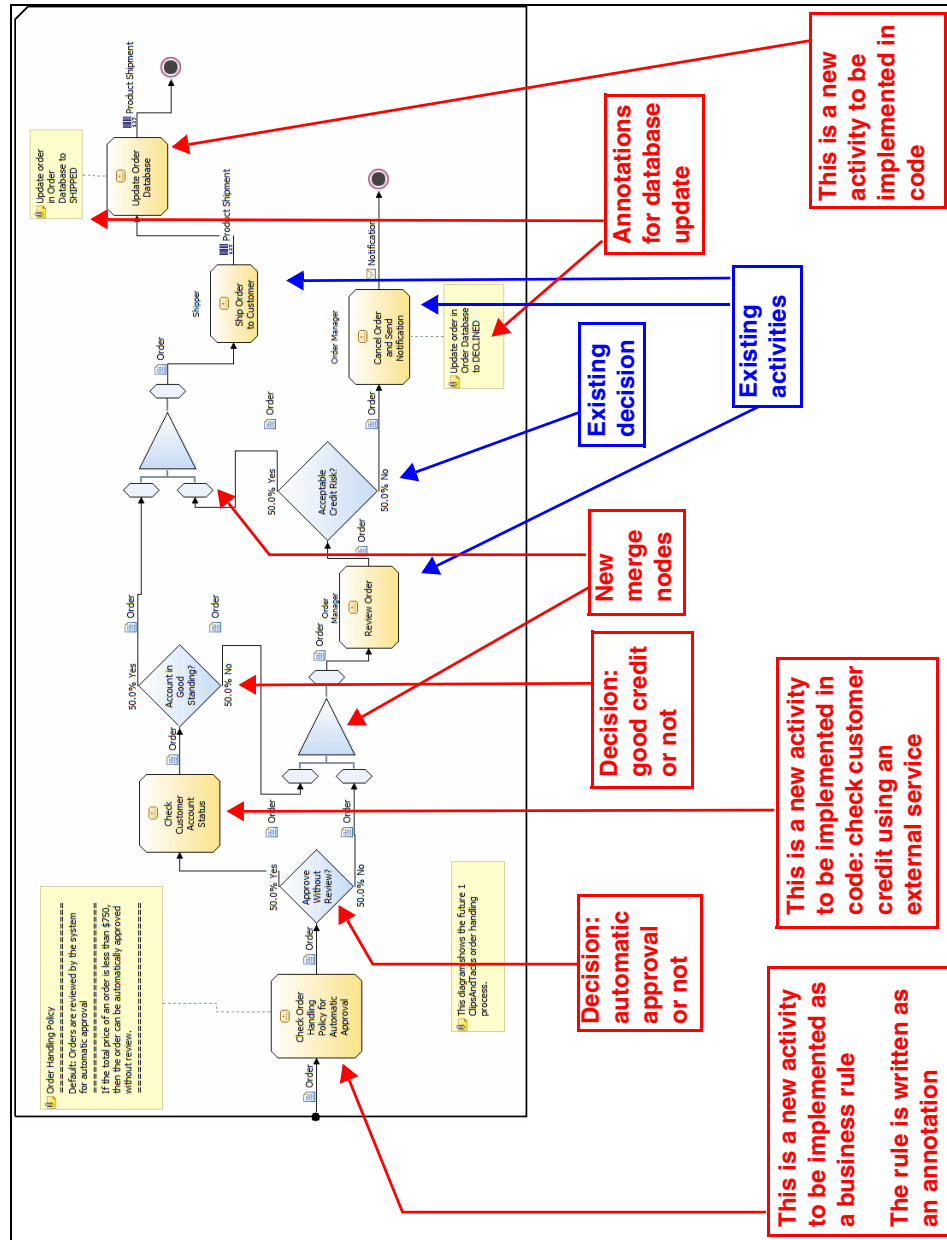


Figure 7-7 Order Handling (Future 1) process diagram

Deleting an activity

Open the Order Handling (Future 1) process.

We transfer the receive order process activities to the customer using a Web application. Select the Receive Order activity and *Delete*.

Creating new activities

Tasks are the basic building blocks representing activities in a process model. Each task performs some function. Visually, a task represents the lowest level of work you can portray in a process.

Tasks are atomic actions, meaning that they cannot be broken down any further, in contrast to processes, which can be decomposed into another flow.

In this section we create three new tasks representing activities in accordance with business analyst requirements:

- ▶ **Check Order Handling Policy for Automatic Approval**—In this activity the ClipsAndTacks company asks to check orders automatically against a business rules engine with the rule shown in Figure 7-8.

```
Order Handling Policy
=====
Default: Orders are reviewed by the system for automatic approval
=====
If the total price of an order is less than $750,
then the order can be automatically approved without review.
=====
```

Figure 7-8 Check Order Handling Policy for Automatic Approval



- ▶ **Check Customer Account Status**—This activity is a back office system activity where the customer credit rating is verified to set an available credit limit. This activity has an insignificant duration.
- ▶ **Update Order Database**—This activity updates the order database after an order has been shipped. It is an automated task, which replaces the current manual task.

Creating an activity

To create an activity, follow the steps outlined below.

Create the activity for automatic approval

Follow these steps:

- ▶ Select the *Create a local task* icon  and move the mouse to the diagram area, and then click to insert the activity. Place the activity where the Receive Order activity was.
- ▶ Select the new activity, and select the *General* tab in the Attributes view.
- ▶ Change the name field to Check Order Handling Policy for Automatic Approval.
- ▶ Define the output path business item (Figure 7-9):
 - Select the *Outputs* tab and click *Add*.
 - Select the column *Associated Data* and click the  icon.
 - Select *Order* and click *OK*.

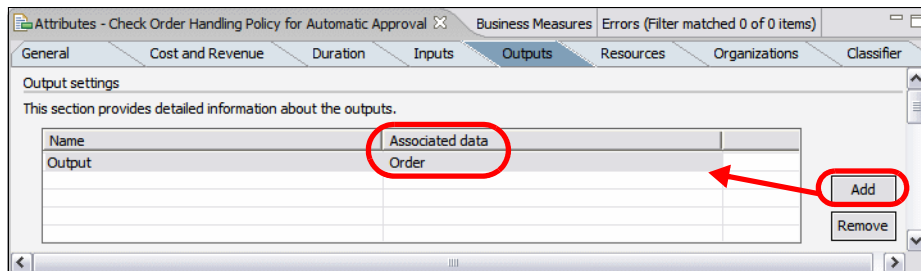


Figure 7-9 Define the output path business item

Create the activity to check the customer account status

Repeat the steps above to create an activity named Check Customer Account Status and define the Order business item as output. Place the activity to the right of Check Order Handling Policy for Automatic Approval with enough space between them to fit in another modeling artefact.

Create the activity to update the order database

Repeat the steps above to create an activity named Update Order Database and define the Product Shipment business item as output. Place the activity to the right of the Ship Order to Customer activity.

Notice that you can also create new tasks (and other objects) by right-clicking in the diagram area and selecting *New* → *Local Task*.

Creating new decisions

A decision routes inputs to one of several alternative outgoing paths. You can think of a decision as a question that determines the exact set of activities to perform during the execution of a process.

Questions might include: “What type of order?”, “How will the order be shipped?”, and “How will the customer pay?”

Decisions are flow control constructs rather than activities such as tasks or processes. They have no costs or duration, and are used to show alternate paths from a preceding activity.

There are two types of decisions: simple decisions and multiple-choice decisions. In this section we use simple decisions.

A simple decision has one incoming branch with one input, and two outgoing branches with one output each. When the process is running, the process flow takes one outgoing branch if a certain condition is true, and the other branch if the same condition is false. The decision selects the outcome based on the incoming data.


Creating the decisions

We create two decision nodes:

- ▶ Approve Without Review?
- ▶ Account in Good Standing?

Create the decision to approve without review

Follow these steps:

- ▶ Select the *Create a simple decision* icon  and move the mouse to the diagram area on the right side of Check Order Handling Policy for Automatic Approval, and then click to insert the decision node (Figure 7-10).
- ▶ Select the decision node, and select the *General* tab in the Attributes view.
- ▶ Change the name field to Approve Without Review?

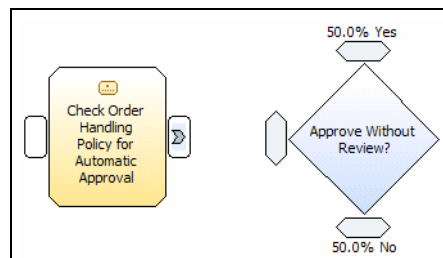


Figure 7-10 Decision Approve Without Review?

Create the decision to verify the account is in good standing

Repeat the steps above and create a decision named Account in Good Standing? on the right side of Check Customer Account Status.

Creating merge nodes

Merges combine multiple processing paths, recombining alternative flows back into a single flow. Merges have multiple incoming branches and one outgoing branch and are used to recombine separate paths in a process flow. A merge is normally used after an exclusive decision. It runs whenever one of its incoming branches is satisfied. As soon as an input is received at a merge, it is immediately sent out as output.

In this section we create two merge nodes necessary to recombine the process flow before the Review Order activity, and to recombine the process flow before the Ship Order to Customer activity.

To create a merge node, select *New* → *Merge* from the diagram's context menu. Place one merge node in front of the Review Order task, and one merge node in front of Ship Order to Customer. Accept the default names (Merge and Merge:2).

Creating connections


A connection is a link between two elements. Connections can be used to specify the chronological sequence of activities in a process. Each task, subprocess, decision, or other element passes control to the next task or element along a connection. You can also associate business items with connections to pass data from element to element. Each connection can only have one associated business item, but you can use multiple connections if you need to pass multiple business items between two elements.

Tip: If you place the mouse pointer over a connection in the process diagram, the source and the target of the connection are displayed.

In this section we connect the new activities, decisions, and merge nodes.

Connect the process input to the automatic approval

Create the connection from the activity process input node to Check Order Handling for Automatic Approval (Figure 7-11).

- ▶ Select the *Create connection* icon .
- ▶ Select the process input node.
- ▶ Move your mouse to the input node of Check Order Handling for Automatic Approval and click.

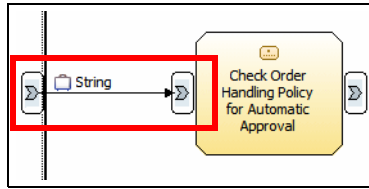


Figure 7-11 Create a connection from the input to Check Order Handling

Create connections between activities, decisions, and merges

Tip: Follow the steps in the exact sequence shown in Figure 7-12. This sequence enables you propagate the output data type (Order).

Follow the steps of the diagram using the connection (arrow) element (Figure 7-12):

1. Select the output node of Check Order Handling for Automatic Approval. Move the mouse to the input node of Approve Without Review? The Yes and No outputs must be grey (that means the data type was set).
2. Select the *No* output of Approve Without Review? Move the mouse to the bottom input of merge node 1.
3. Select the *Yes* output of Approve Without Review? Move the mouse to the input node of Check Customer Account Status.
4. Select the output node of the merge. Move the mouse to the input node of Review Order.

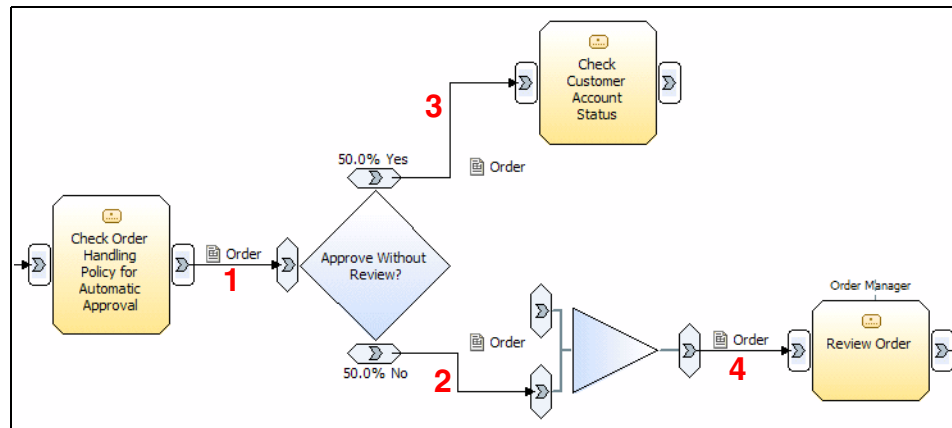


Figure 7-12 Connect and propagate the Order item through the process (step1)

Follow the steps of the diagram (Figure 7-13):

1. Select the output node of Check Customer Account Status. Move the mouse to the input node of Account in Good Standing? The Yes and No outputs must be grey (that means the data type was set)
2. Select the *No* output of Account in Good Standing? Move the mouse to the top input of merge node 1.
3. Select the *Yes* output of Account in Good Standing? Move the mouse to the top input of merge node 2.

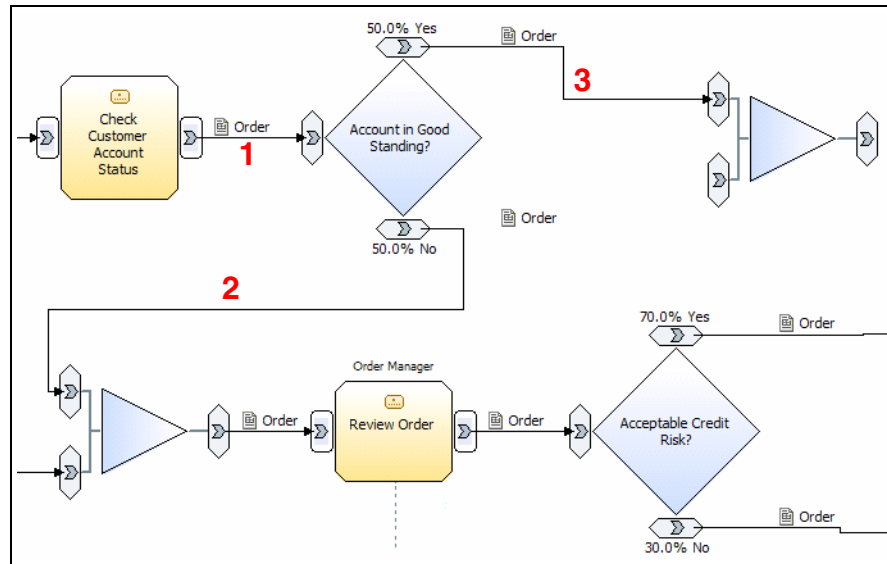


Figure 7-13 Connect and propagate the Order item through the process (step2)

Rewire an existing connection from the decision Acceptable Credit Risk? output to the merge node 2 (Figure 7-14):

1. Select the end of the connection in the input of the activity Ship Order to Customer and move the arrow to the bottom input of the merge node 2.
2. Connect the output of the merge node to Ship Order to Customer.

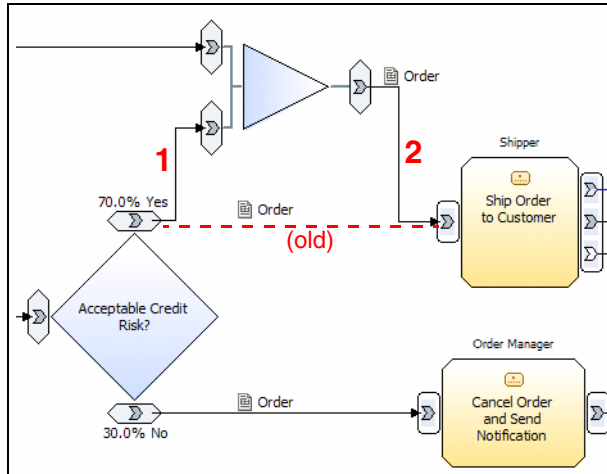


Figure 7-14 Move the connection from Acceptable Credit Risk to the merge node

Rewire the output connection from Ship Order to Customer to the input node of Update Order Database (Figure 7-14):

1. Select the end of the connection going from the output of the activity Ship Order to Customer to the output of the process on the far right of the diagram and move the connection to the input of the Update Order Database activity.
2. Delete the connection going from the output of Ship Order to Customer to the process stop node.
3. Create a connection from the output of Update Order Database to the process stop node.
4. Delete any remaining unwired output nodes from Ship Order to Customer.

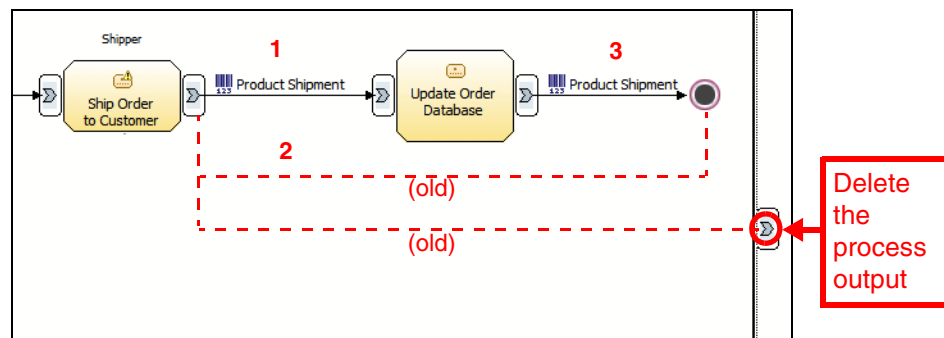


Figure 7-15 Create connections from Ship Order to Customer

Asynchronous process

The Future 1 process is asynchronous. The process ends with a stop node without producing any output. Therefore, select the output arrow at the end of the process (inside the oval shape) and click *Delete* (Figure 7-15 above).

Changing the business item of the process input

Because we deleted the Receive Order subprocess, the business item defined in the input node of the process is wrong. The request item generated by the system is now an order generated by the customer using the Web front-end.

To change the business item of the input process node (Figure 7-16):

1. Select the process input node.
2. In the Attributes view *General* tab, click *Browse*.
3. Select the business item *Order* and click *OK*.

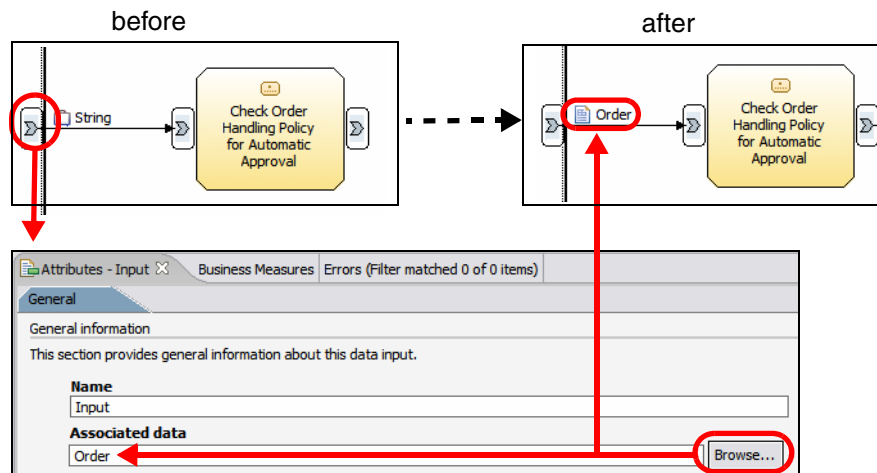



Figure 7-16 Change the business item type of the process input node

Populating resources for the activities

Note: The model available in the sample code does not have any bulk resources, only role requirements (Order Manager and Shipper).

When you model a task, you can specify the individual or bulk resources that are required to complete the task. Resources represent values or instances of resource definitions.

To populate the bulk resources in the Check Order Handling Policy for Automatic Approval activity (Figure 7-17):

1. Select the activity Check Order Handling Policy for Automatic Approval.
2. Select the *Resource* tab, navigate to the Bulk resource requirements area, and click *Add*.
3. Click the  icon and select *Order Management System*.
4. Click *OK*.

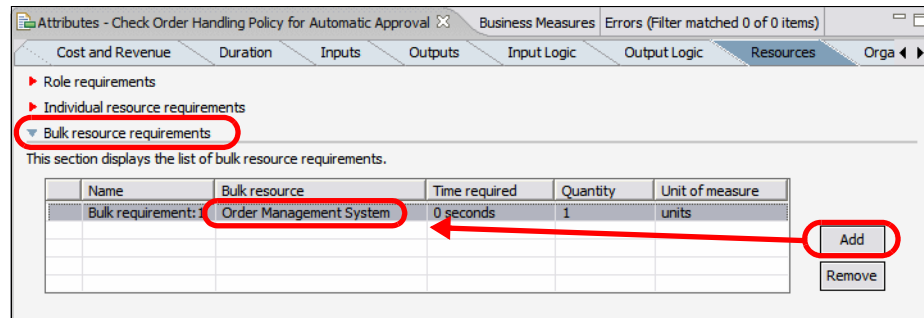


Figure 7-17 Populate resources in Check Order Handling Policy...

Follow the same steps to populate the bulk resources in the Update Order Database activity. For the Check Customer Account Status activity, repeat the steps, but select *Customer Records System*.

Populating expression in decisions

We have to define an expression for every decision node. This element describes the behavior in a runtime mode or a simulation mode instead of probabilities. In our case, this expression is only used in the runtime mode (not in the simulation mode).

Populate the expression for Approve Without Review?

The expression is based on the business analyst description, approve order or send for review, based on the business rule:

- ▶ If the order is under or equal \$750, approve automatically.
- ▶ If the order is over \$750, send for review.

In our case the business rules will be implemented in the Check Order Handling Policy for Automatic Approval activity. So in the decision following the activity, you only have to check the result of the business rules.

- ▶ Select the decision Approve Without Review?
- ▶ Select the *Output Branches* tab and select the *Yes* condition (Figure 7-18).

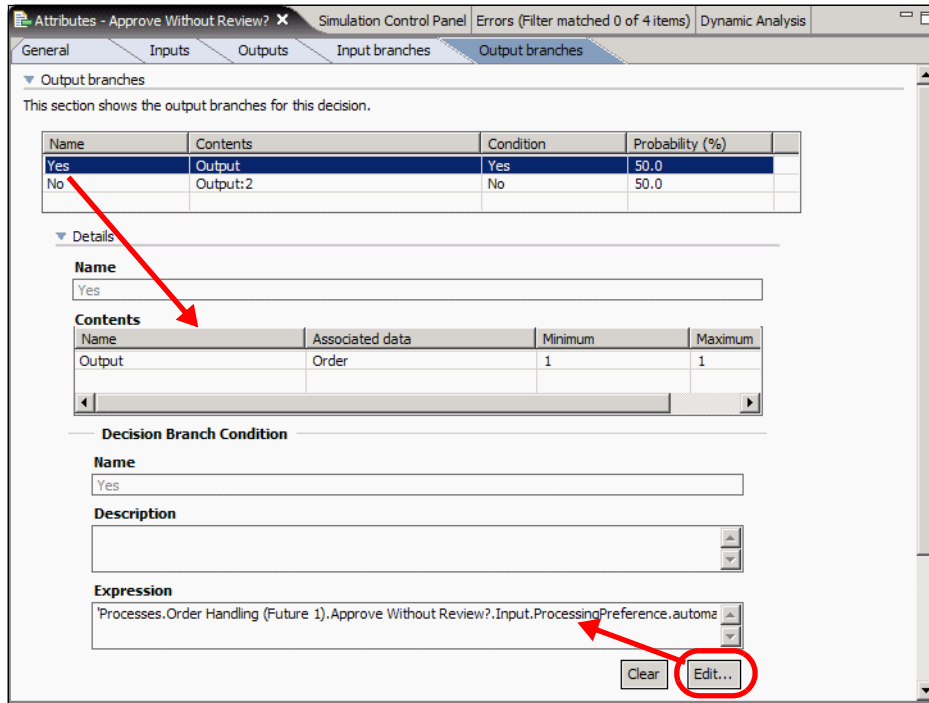


Figure 7-18 Approve without Review decision (Output Branches tab)

- ▶ Navigate to the Decision Branch Condition area and click *Edit*.

Define the expression in the Expression® Builder by following these steps (Figure 7-19):

- ▶ Select the first term type as *Modeling artifact*.
- ▶ Navigate to the element *Processes → Order Handling (Future 1) → Approve Without Review? → Input → Processing Preference → automaticApproval*.
- ▶ Select the operator *is equal to*.
- ▶ Select the second term type as *boolean*.
- ▶ Select the term *true*.
- ▶ Click *Apply*.
- ▶ Click *OK*.

The final expression is:

```
'Processes.Order Handling (Future 1).Approve Without Review?
.Input.ProcessingPreference.automaticApproval' is equal to true
```

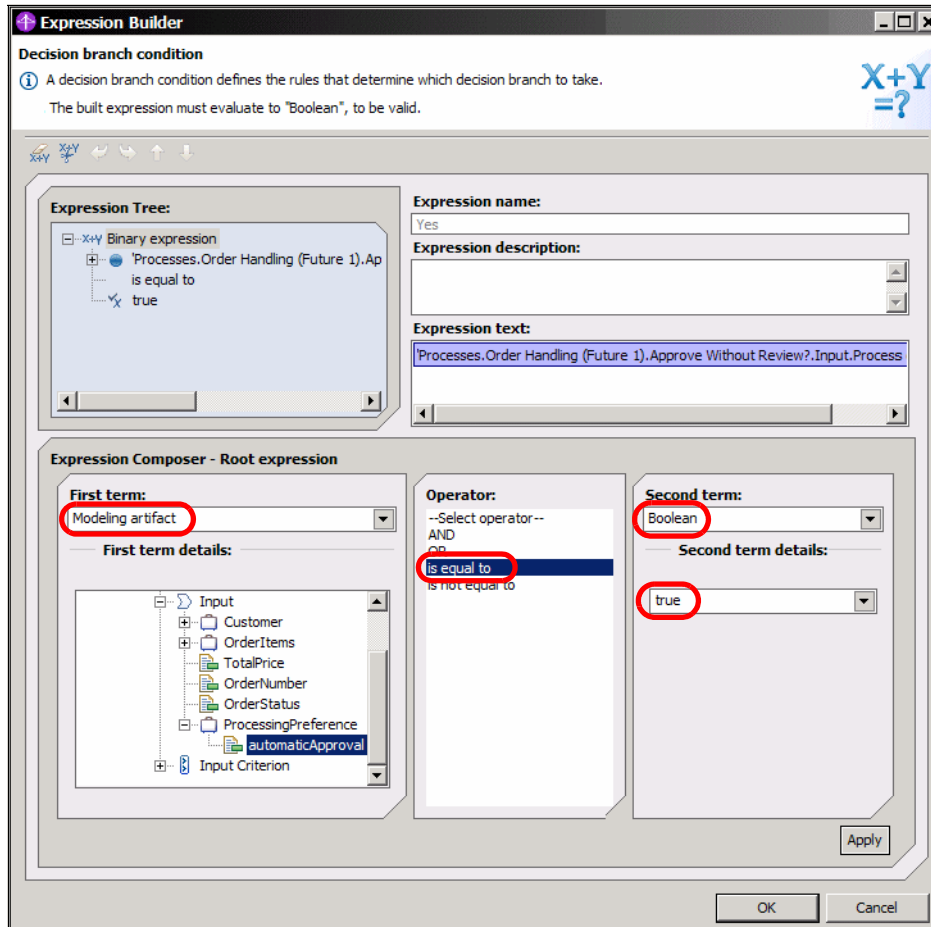


Figure 7-19 Expression Builder: Approve Without Review?

Note: You specified an expression for one of the branches in a simple decision, and WebSphere Business Modeler generates the expression for the other branch for you.

Populate the expression for Account in Good Standing?

Follow the same steps as above, but select these terms and an operator:

- ▶ Select the Account in Good Standing decision, open the Output branches tab, select the *Yes* output branch and drop down to the *Edit* button.
- ▶ Select *Modeling artifact* and navigate to the element *Processes* → *Order Handling (Future 1)* → *Account in Good Standing?* → *Input* → *TotalPrice*.

- ▶ Operator: *is less than or equal to*
- ▶ Select *Modeling artifact* and navigate to the element *Processes → Order Handling (Future 1) → Account in Good Standing? → Input → Customer → AvailableCredit*.

The final expression is:

```
'Processes.Order Handling (Future 1).Account in Good Standing?
.Input.TotalPrice' is less than or equal to '
Processes.Order Handling (Future 1).Account in Good Standing?
.Input.Customer.AvailableCredit'
```

Populate the expression for Acceptable Credit Risk?

Follow the same steps as above, but select these terms and an operator:

- ▶ Select the Acceptable Credit Risk decision, open the Output branches tab, select the *Yes* output branch and drop down to the *Edit* button.
- ▶ Navigate to the element *Processes → Order Handling (Future 1) → Acceptable Credit Risk? → Input → OrderStatus*.
- ▶ Operator: *is equal to*.
- ▶ Select *text* for the second term and enter APPROVED as value.



The final expression is:

```
'Processes.Order Handling (Future 1).Acceptable Credit Risk?
.Input:2.OrderStatus' is equal to "APPROVED"
```

Populating business comments in the process

In the process we defined activities with resources, decisions with expressions, connections with business items. For now the activity behavior is not defined. To communicate the business requirements to the IT community, we have to comment activities where you require a specific behavior.

We provide a comment for the Order Handling Policy for Automatic Approval activity to communicate the business rule (Figure 7-20):

- ▶ Select the *Create annotation* icon  and move the mouse to the diagram area, and click to insert an annotation.
- ▶ Click the *Create annotation connection* icon  and move the mouse to the annotation, click the annotation, and move the mouse to the Order Handling Policy for Automatic Approval activity and click.
- ▶ Enter the annotation text in the diagram or in the Attributes view.

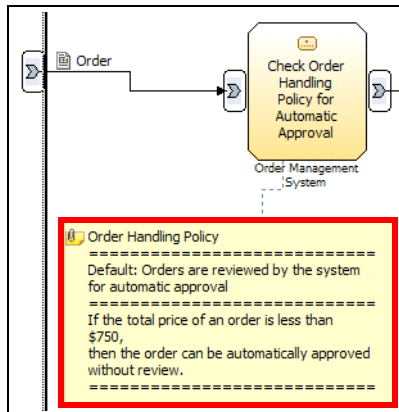


Figure 7-20 Annotation for Check Order Handling Policy for Automatic Approval

- ▶ Also add annotations to the activities Ship Order to Customer

Update order in Order Database to SHIPPED

and Cancel Order and Send Notification

Update order in Order Database to DECLINED

Delete the Order Database that is connected to the Ship Order to Customer activity. We are using the annotation to convey the message to the implementer of the application.

Validating the process

Now the process is complete. To validate the process flow, right-click in the diagram area and select *Static Analysis* → *Paths Unable To Be Followed*. This summary returns a list of the paths within the process that cannot be followed because of an invalid input criterion on an activity in the path.

Creating a valid process is important for simulation and for accurate communication. If there are paths within a model that will never be followed, you have to be aware of this.

If this analysis reveals that the input or output criterion of an activity is modeled in such a way that its path is unable to be followed, you can make changes to the model to ensure that the deficiency is corrected.

The input criterion can be invalid for one of the following reasons:

- ▶ No inputs are specified.
- ▶ One of the inputs of the input criterion does not have an incoming connector (excluding inputs whose input source is a repository or constant value).

If you get an empty summary, your process is fine. If you get an error result as shown in Figure 7-21, navigate in the tree to find an empty element incoming or outgoing, and click the root element to find it in the diagram.

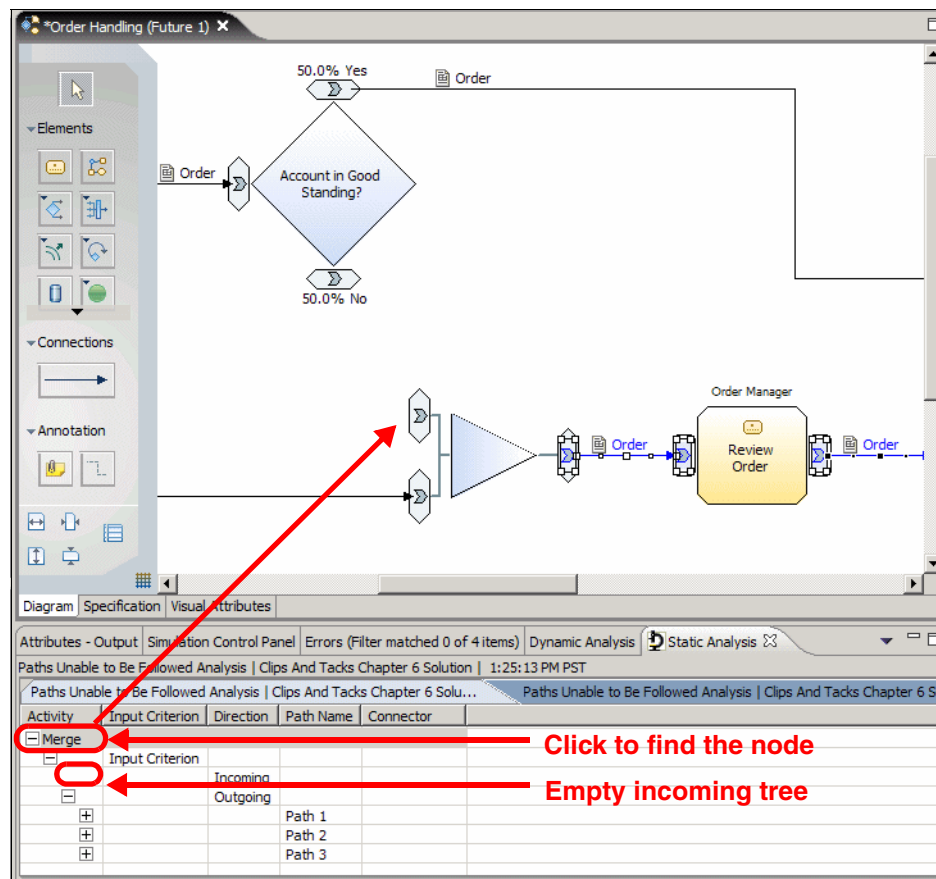


Figure 7-21 Static analysis: Paths unable to be followed result

Organizing the diagram

To organize the diagram in a sequential flow, right-click in the diagram area and select *Auto-Layout Left to Right*. This arranges the layout of the diagram so that the direction of flow goes from left to right, and cleans up any overlapping nodes or connections. The result should be similar to Figure 7-7 on page 130 and Figure 7-22 below.

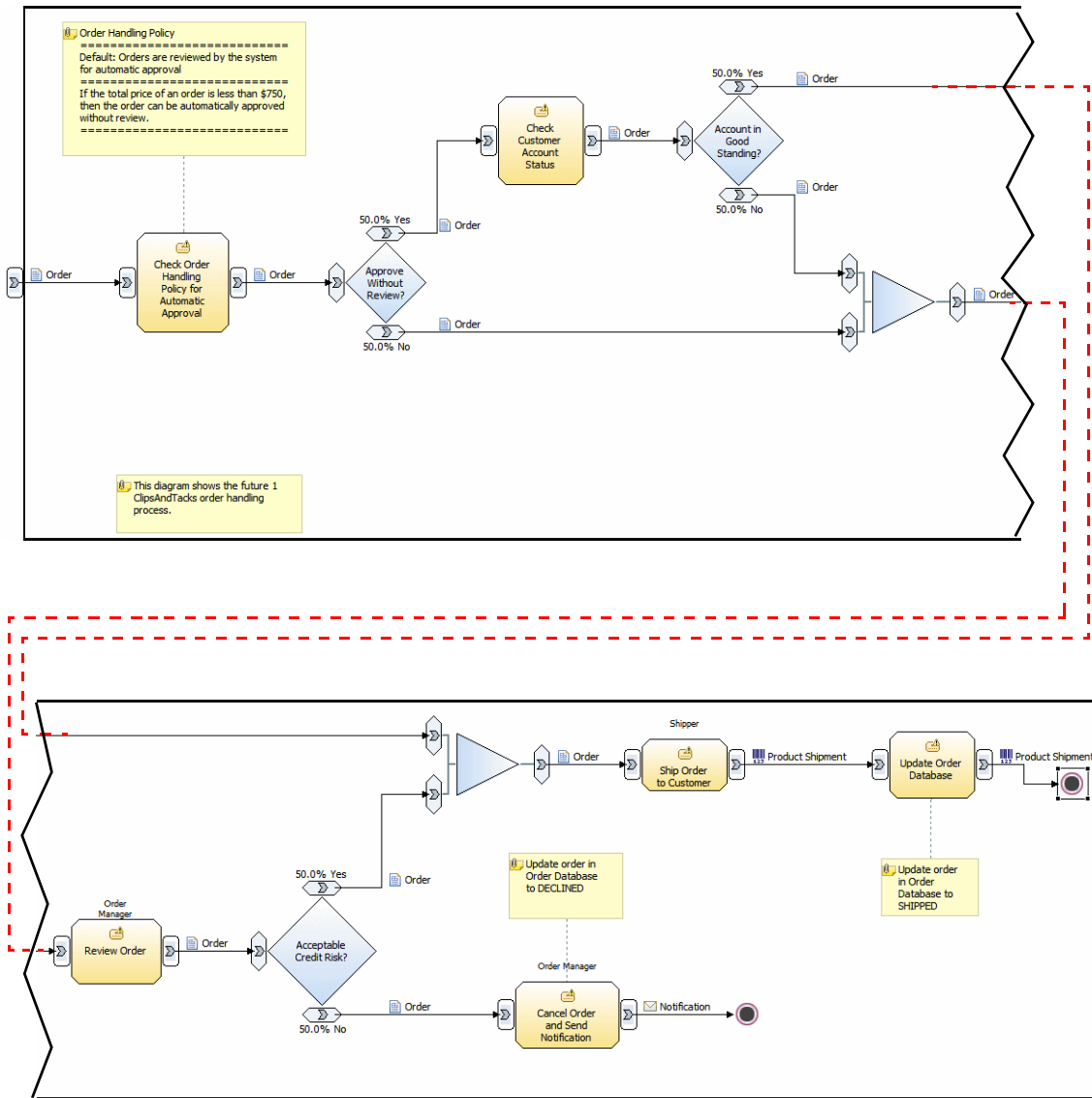


Figure 7-22 Order Handling (Future 1) process diagram

Building an overall process for simulation

To model the customer Web application that provides the input for the Order Handling process, we create a parent process named Overall Order (Future 1) that invokes the Order Handling process.

Creating the main process Overall Order (Future 1)

First we create the new process (Figure 7-23):

- ▶ In the Project Tree, select *Processes* and *New* (context).
- ▶ Enter the name of the process: Overall Order (Future 1).
- ▶ Click *Finish* and the new process opens in the diagram editor.

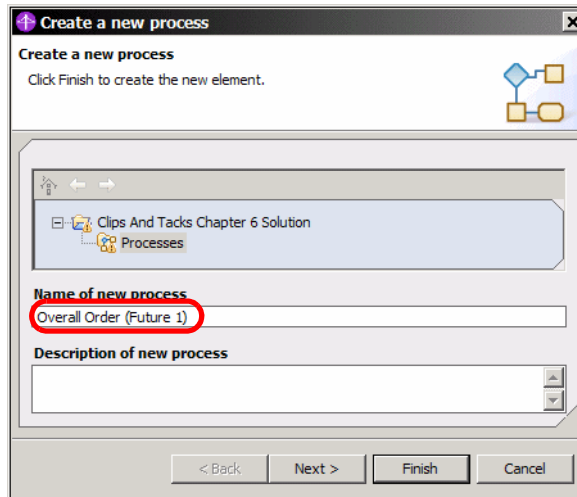



Figure 7-23 Create the new process Overall Order (Future 1)

Create the Receive Order activity

To create the Receive Order activity:

- ▶ Select the *Create a local task* icon  and move the mouse to the diagram area, then click to insert the activity.
- ▶ Select the activity, and in the *General* tab of the Attributes area change the name field to Receive Order.
- ▶ Define the output path in the *Outputs* tab by setting the associated data to the Order item.
- ▶ Define the input path in the *Inputs* tab by setting the associated data to the Request item.

Create the subordinate process Order Handling (Future 1)

To create the subprocess call to Order Handling (Future 1):

- ▶ Right-click the diagram area and select *New* → *Global Process*.
- ▶ Select the Order Handling (Future 1) process.

Create a Stop node in the process

Click the  icon and select the *Stop* icon  and drop it into the diagram.

Delete the Start node and the existing Stop node (very far at bottom right).

Create the connections

Connect the activities in sequential way:

- ▶ Input node to the input of the Receive Order activity.
- ▶ Output of the Receive Order activity to the input node of the Order Handling (Future 1) process.
- ▶ Output of the Order Handling (Future 1) process to the Stop node.
- ▶ The finished process is shown in Figure 7-24.

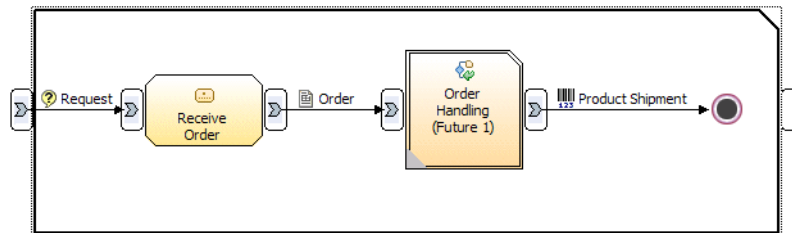




Figure 7-24 Overall Order (Future 1) process

Populate the bulk resource in the Receive Order activity

We define Customer and Web Application as resources (Figure 7-25):

- ▶ Select the activity Receive Order.
- ▶ In the *Resource* tab, navigate to the Bulk resource requirements area, and click *Add*.
- ▶ Click , select *Customer*, and click *OK*.
- ▶ Click *Add*, click , select *Web Application*, and click *OK*.

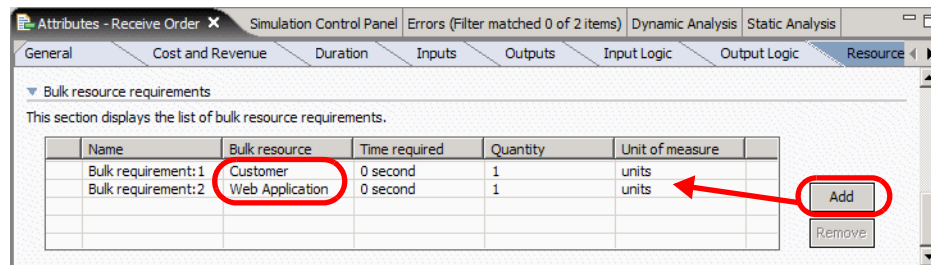


Figure 7-25 Populate resources in receive order activity

Business Process Execution Language

We can export files that can be imported into WebSphere Integration Developer to create an implementation for WebSphere Process Server. We can select an entire project, a data catalog, process catalog, resource catalog, or organization catalog with its contents, or a single process, global task, service, or business item. If you select a process, that process and any business items it references are also exported.

You can export a project as a set of Business Process Execution Language (BPEL) files for exporting into WebSphere Integration Developer. Integration Developer is the tool used to implement business processes for deployment on Process Server. In Integration Developer you assemble the application by tying the activities defined in the business process to implementations, which specify how the activities are accomplished. These implementations can be existing IT assets such as reusable Web services, access to databases and EIS, where customer accounts and credit information are typically stored, or other business processes executing in BPEL, thus building composite applications. However, often defining new business processes will result in the requirement for new IT assets, which the IT department will then need to create.

Finally, you can test the application and deploy the application to Process Server.

Preparing the process for export

Before exporting a project destined for WebSphere Process Server, ensure that you are working in the WebSphere Process Server mode and that your model has no errors.

To set the Modeler into WebSphere Process Server mode, select *Modeling* → *Mode* → *WebSphere Process Server*.

The exported process does not necessarily guarantee a complete process definition. For example, if the decision branches in the model have no formal expressions associated with them, no BPEL transition conditions will be generated. In addition, there are certain BPEL export restrictions that prevent all models from being fully transformed.

Warnings

As soon as you switch to WebSphere Process Server mode, warnings show up in the Project Tree and Errors view (Figure 7-26):

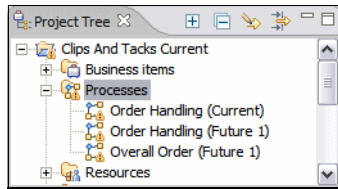


Figure 7-26 Project Tree with warnings in Process Server mode

- ▶ Bulk resources are not supported.

Before exporting the model, you have to change some elements to be compliant with the WebSphere Process Server mode. Open the Order Handling (Future 1) process flow.

Populate technical attributes

To transfer information between the business analyst and the integration designer using WebSphere Integration Developer, the analysts can add some notes in the process diagram that will be transferred through the BPEL export.

Define a comment for the Ship Order to Customer activity (Update order in Order Database to SHIPPED) and Cancel Order and Send Notification activity (Update order in Order Database to DECLINED). Note that we already did this in “Populating business comments in the process” on page 142.

Business process integration and automation (BPIA) project

The Zurich Research Lab is developing tools (Eclipse plug-ins) to bridge the gap between business-level and IT-level models:

<http://www.zurich.ibm.com/csc/bit/bpia.html>

Today's business applications and their underlying process models are becoming more and more complicated, making the implementation of these processes an increasingly challenging task. On the one hand, tools and methods exist to describe the business processes. On the other hand, different tools and methods exist to describe the IT artifacts implementing them. But a significant gap exists between the two. To overcome this gap, new methodologies are sought. Our research combines three recent trends in the IT industry: OMG's Model-Driven Architecture, Web services, and BPEL.

Defining the type of implementation for all activities

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

The technical attributes define a set of component properties including the implementation type of the component. The Modeler provides six types of implementation:

- ▶ Rule group (to be implemented as a business rule)
- ▶ Human task (human interaction using Web dialogs)
- ▶ Java
- ▶ Process (to be detailed at implementation time and may decompose)
- ▶ State machine (not used in our examples)
- ▶ Import with Web Service binding, SCA binding, or JMS binding (new 6.0.2)

If the technical attributes specify an implementation type, the Modeler generates the default implementation based on the implementation type. The default implementation provides a starting point to implement the component in Integration Developer. The implementation type can also be specified by the Integration Developer at assembly time in Integration Developer.

Rule group

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. An example of a simple rule group is shown in Figure 7-27.

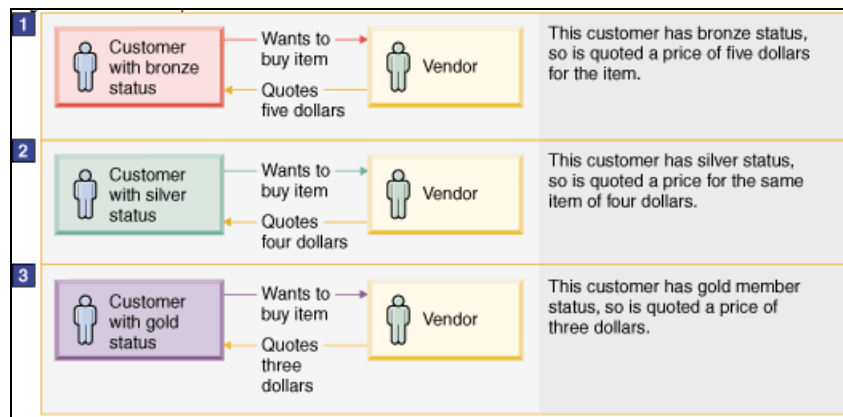


Figure 7-27 Example of a simple business rule group

The Integration Developer tools have been designed so that users can easily compose integrative business solutions without programming skills. To this end, you can easily create and develop business rules in an intuitive graphical programming environment (Figure 7-28).

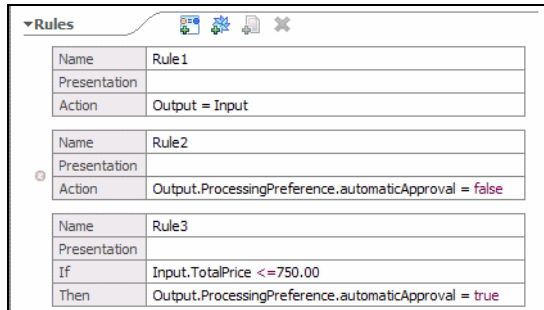


Figure 7-28 Activity rule group implementation

We will implement the task Check Order Handling Policy for Automatic Approval as a rule group:

- ▶ Select the Check Order Handling Policy for Automatic Approval activity in the process flow.
- ▶ In the Technical Attributes view select the *Implementation* tab (Figure 7-29).
- ▶ Select *Rule Group* in the Implementation type.

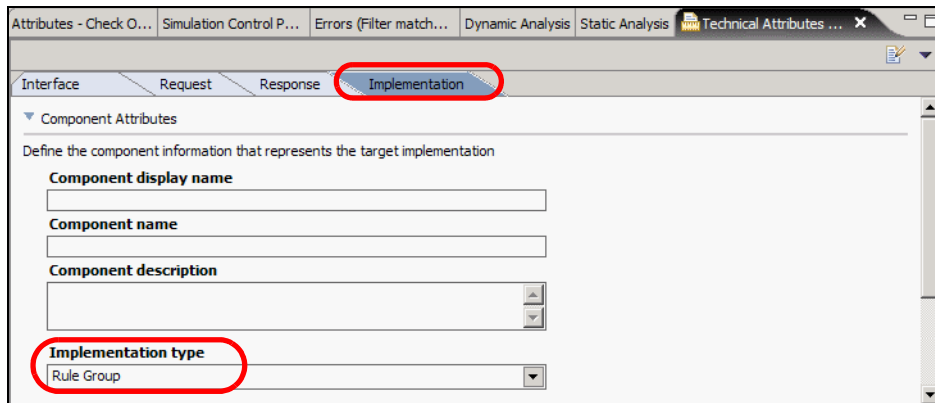


Figure 7-29 Check Order Handling Policy: Implementation type

Human tasks

A human task is, quite simply, a unit of work done by a human. Quite often, this task involves the interaction with other services, and thus becomes a task within a larger business goal.

Examples of human tasks are shown in Figure 7-30.

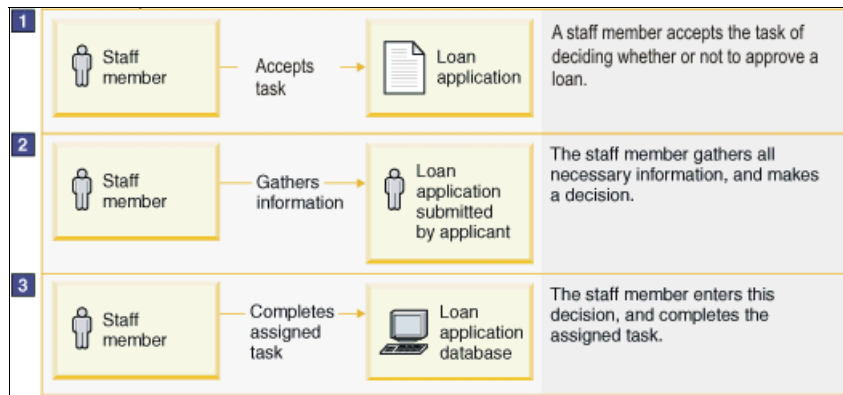


Figure 7-30 Examples of human tasks

The Integration Developer tools have been designed so that users can easily compose integrative business solutions without programming skills. To this end, you can easily create and develop a human task in an intuitive graphical environment called the human task editor (Figure 7-31).

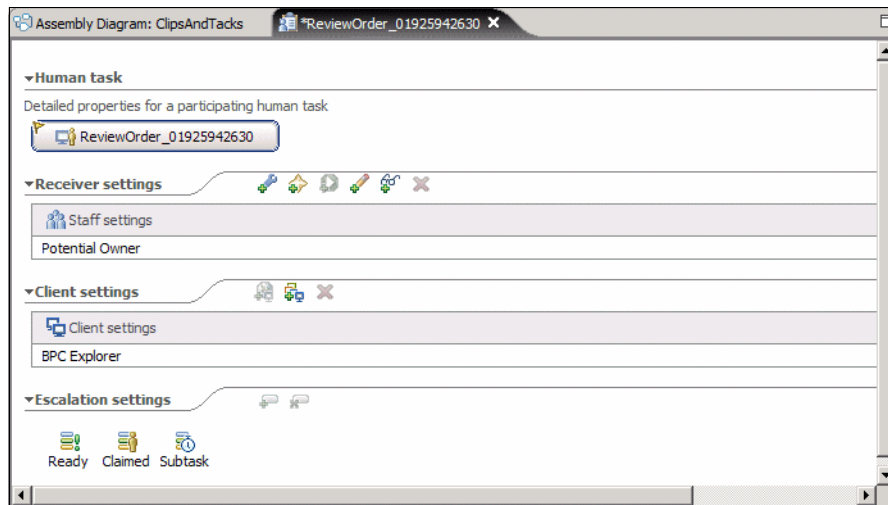


Figure 7-31 Activity human task implementation

The Modeler export generates two types of human tasks: inline task, which is similar to the Staff activity generation of WebSphere Business Integration Server Foundation V5, and a human task component, which is an SCA component that has the implementation type of human task:

- ▶ To export as an inline task, the task must have a role requirement or an individual resource requirement. 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 an activity as a human task component, the implementation type must be set to human task in the technical attributes.

We implement the activities Review Order and Ship Order to Customer as human tasks.

Repeat the next steps for each activity:

- ▶ Select the activity, for example, Review Order.
- ▶ In the Technical Attributes view, select the *Implementation* tab.
- ▶ Select *Human Task* for the Implementation type (similar to Figure 7-29 on page 151).

Java

The Java implementation type allows you to create the activity using the Java language.

In Integration Developer, we will write Java code to implement the activity:

```
public DataObject InputCriterion(DataObject Input) {
    System.out.println("Check Customer Account Status Invoked");

    // write Java code
    // minimally copy the input data object as output data object
    DataObject Output = Input;
    return Output;
}
```

We will implement the Check Customer Account Status, Cancel Order and Send Notification, and Update Order Database activities as Java tasks:

- ▶ Select the activity, Check Customer Account Status.
- ▶ In the Technical Attributes view, select the *Implementation* tab.
- ▶ Select *Java* for the Implementation type (similar to Figure 7-29 on page 151).

Repeat these steps for Cancel Order and Send Notification and Update Order Database.

Process

A component can be implemented as a process. On export the component becomes a BPEL process.

The Integration Developer provides the necessary tools to easily create and develop a process task in an intuitive graphical programming environment called the process editor (Figure 7-32).

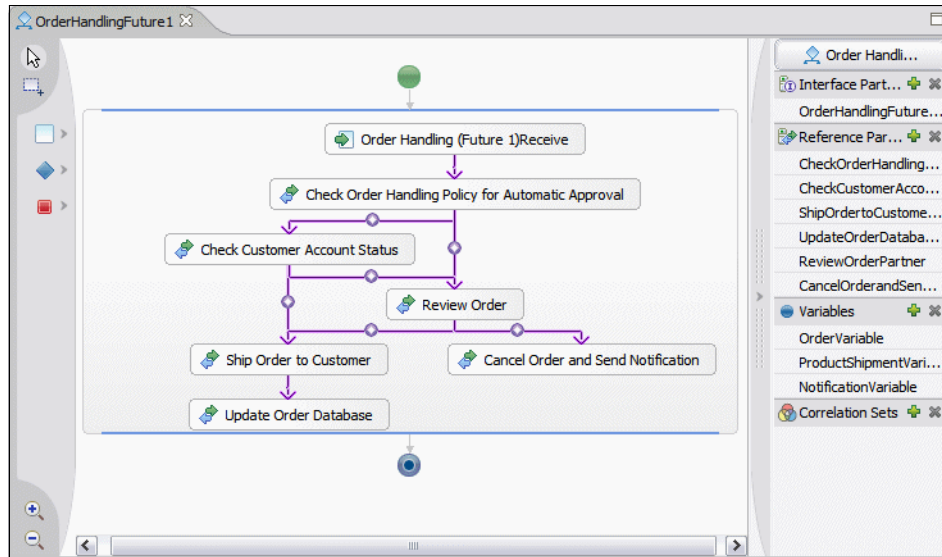


Figure 7-32 Component with an implementation type of process

Note: The business process itself is exported as BPEL.

State machine

A state machine is an event driven business transaction in which external operations trigger changes that guide the transaction from one discrete mode to another. Each mode is an individual state, and this mode determines what activities and operations can occur.

The Integration Developer tools have been designed so that users can easily compose integrative business solutions without programming skills. To this end, you can easily create and develop business state machines in an intuitive graphical programming environment called the business state machine editor (Figure 7-33).

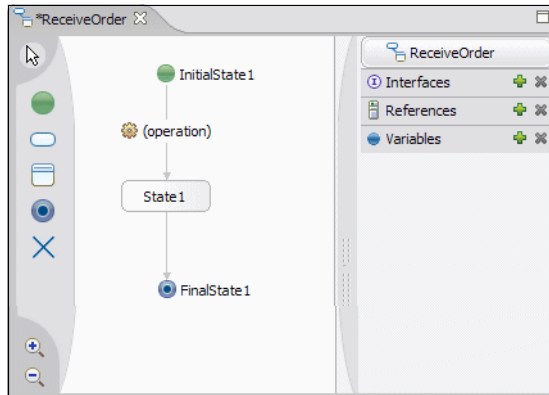


Figure 7-33 Activity state machine implementation

Figure 7-34 shows an example of a state machine.

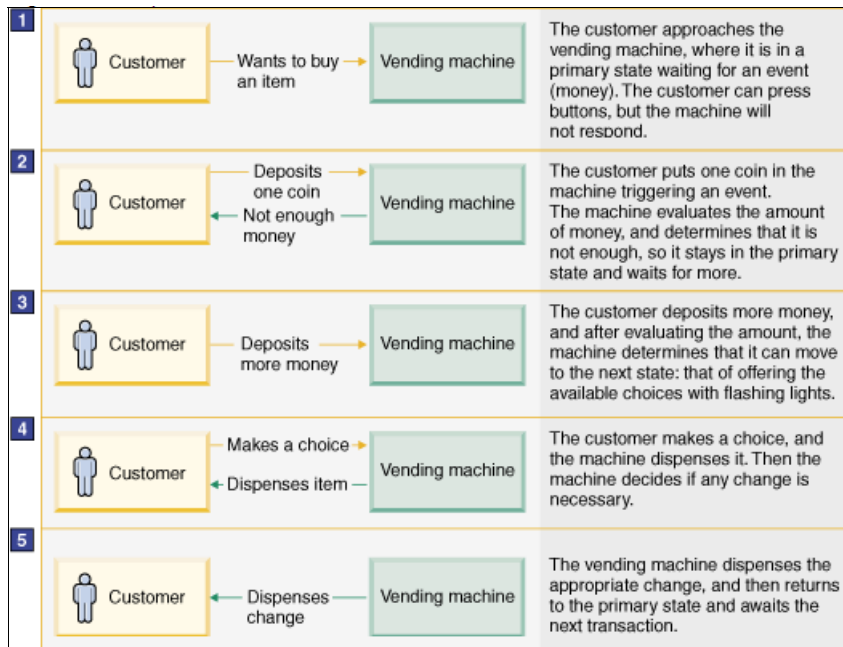


Figure 7-34 Example of a simple state machine

In our example, we do not use an implementation using a state machine.

Import

For tasks and services, the implementation type can also be defined as an SCA Import with Web service binding, SCA binding, or JMS binding. Model elements with these implementation types appear in the Assembly diagram in Integration Developer as Imports with the specified binding.

For business service operations (imported WSDL operations) that are included in a process, WebSphere Business Modeler exports them as an SCA Import with Web service binding.

Note: We will use an imported Web service in our Future 2 model as described in “Creating the Future 2 process in the Modeler” on page 496.

Defining the operation type for each activity

By default the Modeler assigns an operation type of Request/response. This is suitable for all the activities in our model.

If any activity is a one way operation, you would set the operation type to *One way operation* in the Technical Attributes view (Figure 7-35).

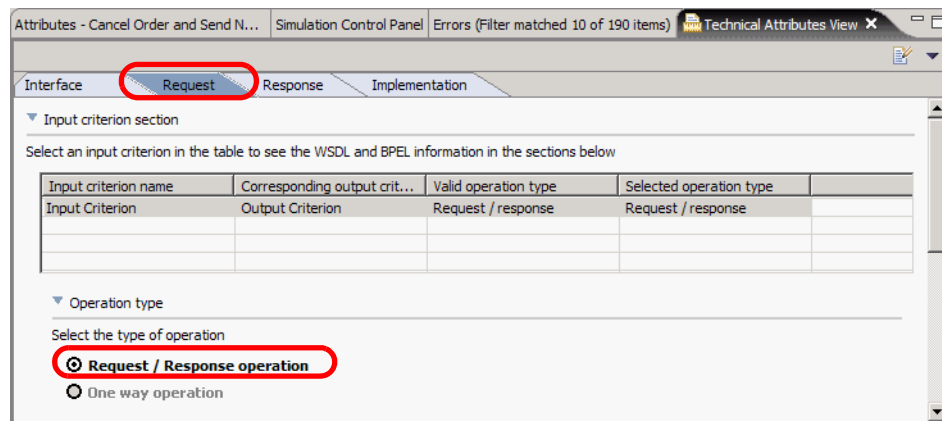


Figure 7-35 Setting the operation type

Defining the operation type for the process

Our process is a one way operation. Once an order is submitted no result is returned. Select the process in the diagram, then select the *Input Criterion* in the Attributes view and set the operation to *One way operation* (Figure 7-36).

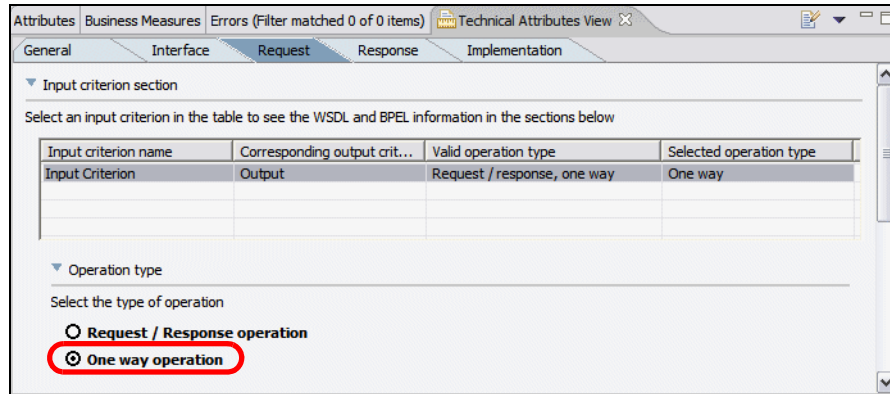


Figure 7-36 Setting the operation type for the process

Exporting the BPEL for WebSphere Integration Developer

After all export parameters are set, we can export the process into BPEL format, to be imported into Integration Developer.

Note: In Chapter 9, “Defining KPIs and measures” on page 187 we enhance the model with business measures to be used by WebSphere Business Monitor to measure and validate the requirements of the business process.

The export of the model for Integration Developer and Process Server is independent of the business measure that we may define. We can export the model now and use it in Chapter 10, “Developing the application using WebSphere Integration Developer” on page 211.

If you want to work **with Business Monitor**, you can first go through Chapter 9, “Defining KPIs and measures” on page 187 before exporting the process.

Exporting the model for Integration Developer

To export the model as BPEL, perform these steps:

- ▶ Select the ClipsAndTacks project and select *Export* (context).
- ▶ Then, select the type of export as *WebSphere Process Server* export (Figure 7-37).

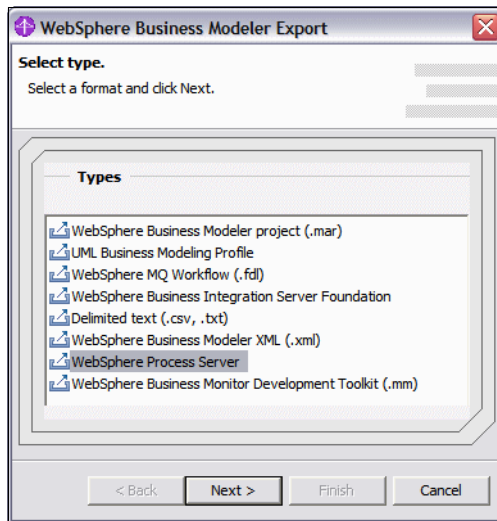


Figure 7-37 WebSphere Process Server export

Now, we have to define how we want to export:

- ▶ If you specified a module or library project, the export process creates a folder that contains all of the project's files.
- ▶ If you further specify a project interchange name, the export packages the folder into a ZIP file. The result will be a ZIP file in interchange format that can be imported into Integration Developer as is.
- ▶ If you do not specify a module or library, the export process creates a set of folders with BPEL and WSDL files for the exported processes and XSD files for the business items used within the processes. The files are placed into sub folders based on the catalog hierarchy of the process and business items.

Tip: Module and library:

- ▶ A module without a library produces one base project with all the definitions (process, activities, data types, WSDL interfaces).
- ▶ A module with a library places the shared components into the library project. This includes the data types and the WSDL interfaces. The process and the activities remain in the module.

If you want to share data types and interfaces for multiple business processes, then select module and library.

The export of the ClipsAndTacks example is shown in Figure 7-38.

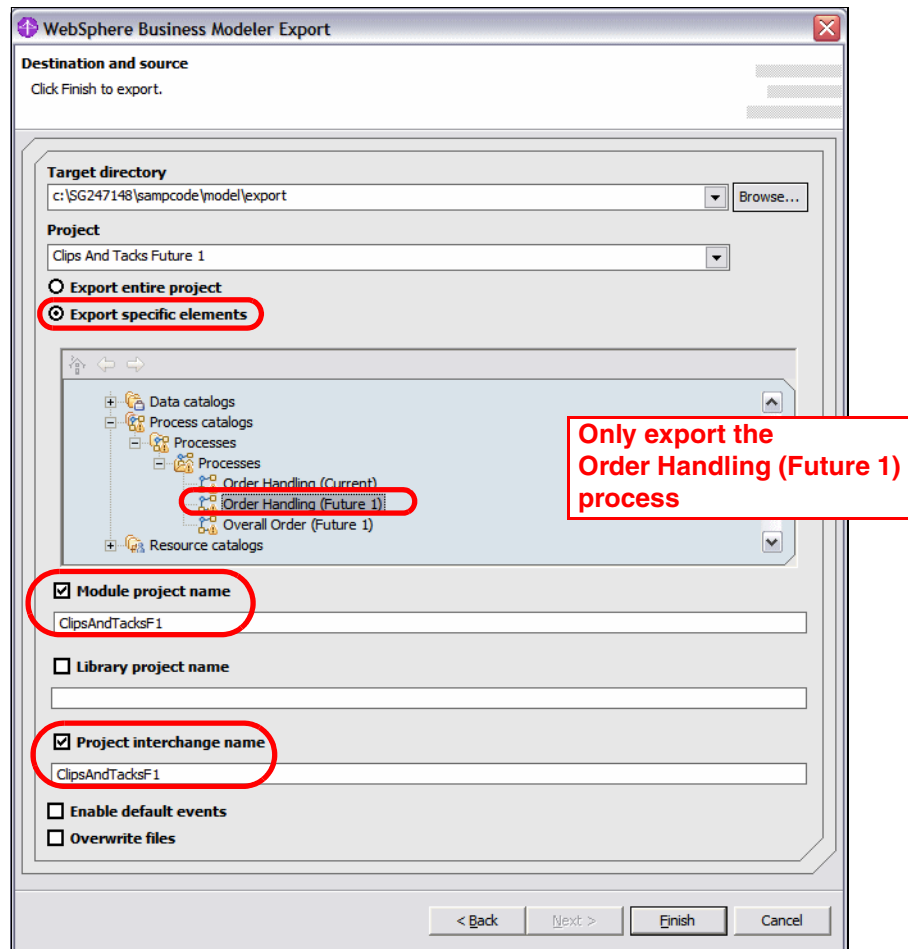


Figure 7-38 WebSphere Process Server export dialog

- ▶ Enter the target directory, for example:
c:\SG247148\samcode\model\export
- ▶ The project is preselected.
- ▶ Select *Module project name* and enter the names as **ClipsAndTacksF1**.
This name, with a suffix of App, becomes the name of the enterprise application in Integration Developer and Process Server.
- ▶ Select *Project interchange name* and enter the name as **ClipsAndTacksF1**.
This name does not have to match the project name.

- ▶ Select *Export specific elements* and navigate to the process Order Handling (Future 1). Select the process.
You do not have to select the business items; they are exported automatically with the selected process.
- ▶ Do not select *Enable default events*. This check box will generate events for all the activities. We rather delay event generation to the implementation in Integration Developer and select only the activities that are relevant for monitoring.
- ▶ Click *Finish*.

Restriction: If you have processes with errors (in WebSphere Process Server modeling mode) you cannot export them to Integration Developer. In this case you have to export selected elements (processes) that are free of errors.

Exported file and content

The exported project interchange file (ClipsAndTackF1.zip) contains all the elements required for Integration Developer. Inside the zip file, you find BPEL and WSDL files organized by process folders (Figure 7-39):

- ▶ There is one .bpe1 file for the business process.
- ▶ For each activity, there is a .component and a .wsdl file. The .component file defines the service component and the .wsdl file defines its interface.
- ▶ For the rule group activity, there is a .brg and a .brgt file. These files define that the component must be implemented using a business rule.
- ▶ For the Java activity, there is a .java file. This file defines a Java class with the required methods. One of the methods must be completed in the development tool.
- ▶ For human tasks, there is a .tel file. This file defines the component as a human task.
- ▶ For the process and the human tasks, .mon files enabling Monitor events are generated if you select *Enable default events* in the export.

Notes:

- ▶ All spaces and special characters are removed from the generated names.
- ▶ The numbered suffixes may be different in your case. The numbers change when you copy or rename a process.
- ▶ The BusinessItems.xsd file contains all the business items used in the process.

```

ClipsAndTacksF1
  .project
  sca.module
  sca.modulex
  businessitems
    Businessitems.xsd
  processes
    orderhandlingfuture1
      CancelOrderandSendNotification1177120723Interface.wsdl
      CancelOrderandSendNotification_01780893802.component
      CancelOrderandSendNotification_01780893802Impl.java
      CheckCustomerAccountStatus0319499984Interface.wsdl
      CheckCustomerAccountStatus_1292162843.component
      CheckCustomerAccountStatus_1292162843Impl.java
      CheckOrderHandlingPolicyforAutomaticApproval.....Interface.wsdl
      CheckOrderHandlingPolicyforAutomaticApproval_517466816.brg
      CheckOrderHandlingPolicyforAutomaticApproval_517466816.brgt
      CheckOrderHandlingPolicyforAutomaticApproval_517466816.component
      OrderHandlingFuture1.bpel
      OrderHandlingFuture1_1581013419.component
      OrderHandlingFuture1_1581013419.export
      OrderHandlingFuture1Artifacts.wsdl
      OrderHandlingFuture1Interface.wsdl
      ReviewOrder0129443853Interface.wsdl
      ReviewOrder_01925942630.component
      ReviewOrder_01925942630.tel
      ShipOrdertoCustomer01971481858Interface.wsdl
      ShipOrdertoCustomer_0567294367.component
      ShipOrdertoCustomer_0567294367.tel
      UpdateOrderDatabase0764662423Interface.wsdl
      UpdateOrderDatabase_164806352.component
      UpdateOrderDatabase_164806352Impl.java

      OrderHandlingFuture1_bpel.mon (if Enable default events selected)
      ReviewOrder_01925942630_tel.mon
      ShipOrdertoCustomer_0567294367_tel.mon

```

Figure 7-39 Structure of exported project files

We will use the interchange file in Chapter 10, “Developing the application using WebSphere Integration Developer” on page 211.

Summary

In this chapter we described how to implement changes in a business process based on simulation and analysis.

We then described how to set up implementation details for WebSphere Process Server and how to export the model into a file suitable for WebSphere Integration Developer.



Simulating and analyzing the Future 1 process

This chapter describes how the ClipsAndTacks Order Handling (Future 1) process was simulated and analyzed once it was revised in the Modeler.

First, we recompile all the Future 1 process model information related to the simulation, then we enter this information into the simulation attributes of the model and generate simulation snapshots.

Once simulation results and statistics are available, we analyze the new simulation performance statistics from the Modeler analysis reports and make conclusions if the new revised process meets the objectives set by the company management.

Overview of simulating the Future 1 process

For the simulation and analysis of the Future 1 process, we are going to perform the same steps of the process simulation as we described in Chapter 6, “Simulating and analyzing the current process” on page 91:

- ▶ Define resource requirements and decision probabilities:
 - Review corporate strategy and objectives
 - Review the Future 1 process model
 - Define these matrixes:
 - Roles and costs
 - Activity durations
 - Resource availability
 - Probabilities on process decisions
- ▶ Define the simulation profile and attributes related to the simulation run
- ▶ Enter the simulation attributes in the Modeler
- ▶ Create and run a simulation snapshot
- ▶ Analyze simulation results
- ▶ Conclusion

ClipsAndTacks process assessment for the simulation

Following the process simulation methodology outlined above, the lead business analyst has completed the following steps:

- ▶ **Review corporate strategy and objectives**

The lead business analyst has confirmed that company objectives for the revised process are as follows:

 - The high-level business objectives are to increase revenue and reduce costs.
 - Specifically, management wants to achieve these objectives:
 - Reduce the average time from when orders are received to the time they are shipped to 3 days.
 - Achieve an order shipping rate of 90% or better.

► **Review the Future 1 process model**

The revised order handling process model is shown in Figure 8-1, which is a compressed version of Figure 7-22 on page 145. The business analyst has reviewed all the process model components to prepare for the next step of defining the simulation related matrixes.

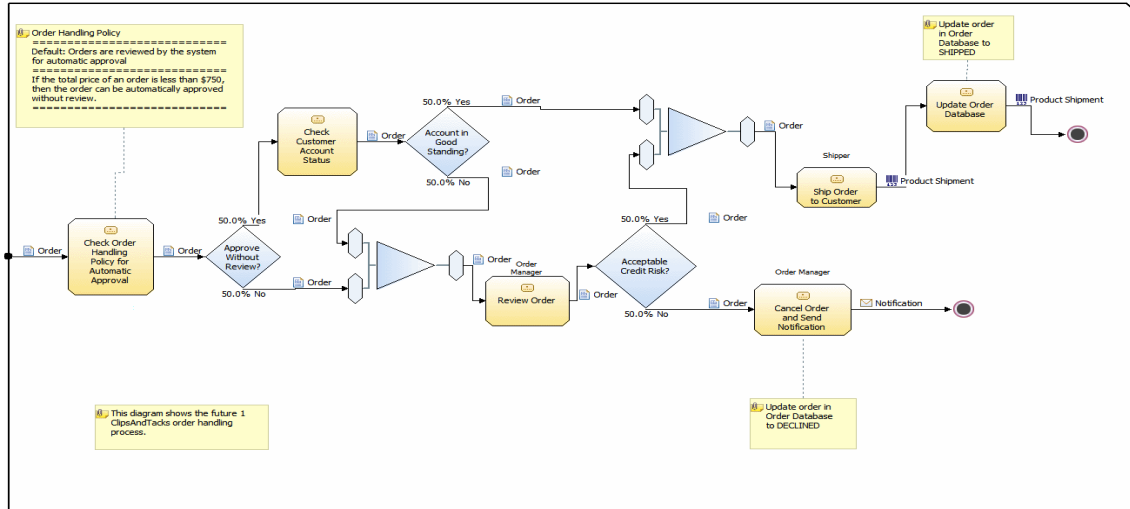


Figure 8-1 Order Handling (Future 1) process model

As a result of the revision of the future process, the following matrixes are compiled:

- Roles and costs
- Activity durations
- Resource availability
- Probabilities on process decisions

Roles and costs matrix

The roles and costs matrix (Table 8-1) shows roles with the assignment of cost to activities. The cost for human tasks is defined by the salary divided by the unit of measure, an hour in our case. The differences from the current process model are as follows:

- Customer Services Representative role is removed.
- Customer and Order Management System roles are added.

Table 8-1 Roles matrix: Role and cost per activity

Activities \ Roles	Cost \$	Receive Order	Check Order Handling Policy	Check Customer Account Status	Review Order	Cancel Order and Send Notification	Ship Order to Customer	Update Order Database
Customer	0	X						
Order Manager	20				X			
Shipper	10						X	

Duration matrix

The duration matrix (Table 8-2) shows the durations of human or system tasks for a specific role and activity. In this example, there is only one role per activity, however, there could be multiple human roles for one activity. The differences from the current process model are as follows:

- ▶ Customer Services Representative role is removed, Customer, Customer Records System, and Order Management System roles and corresponding activity durations are added.
- ▶ Order Manager time is reduced to 15 minutes (better training).

Table 8-2 Duration matrix: Duration of activity per resource

Activities \ Resources	Receive Order	Check Order Handling Policy	Check Customer Account Status	Review Order	Cancel Order and Send Notification	Ship Order to Customer	Update Order Database
Activity duration	7 min 20 sec	1 sec	5 sec	15 min	1 sec	16 min	1 sec
Customer	7 min 20 sec						
Web application	30 sec						
Order Manager				15 min			
Shipper						15 min	

Availability matrix

The resources availability matrix (Figure 8-2) shows the timetables assigned to the resources. In our business case there are two timetables used for the Future 1 process model. The timetables are defined in the modeler as follows:

- ▶ Day Shift
 - There are 9 working hours a day.
 - Working days are Monday to Friday.
 - Working hours are 8:00 AM to 5:00 PM.
- ▶ Online Application
 - 24 hours x 7 days a week

Resources/ Timetable	Day Shift	Online Application
Web Application		
Order Manager		
Shipper		

Figure 8-2 Availability matrix: Timetable per resource

Probabilities on process decisions

The decision probability matrix was updated for the Future 1 process as shown in Figure 8-3.

Decision / Probability	Yes	No
Acceptable Credit Risk	70%	30%
Approve without Review	65%	35%
Account in Good Standing	85%	15%

Figure 8-3 Decision matrix: yes-no probability per decision

Populating the simulation environment

We have to enter all the required information for process simulation in the Modeler.

Note: If you imported the model, all the simulation details are already defined.

Entering simulation attributes in the Modeler

All the required simulation information has been defined and we are ready to enter these simulation attributes in the Modeler. The step-by-step details for entering simulation related attributes into the Modeler are described in “Populating the simulation environment” on page 96.

In this chapter we show the results of the model population with the simulation attributes.

The following tasks have to be completed:

- ▶ Populate role resources with costs and availability
- ▶ Populate activity duration for all activities
- ▶ Populate probability on the output branches for all decisions

Populating role resources with costs and availability

The main difference in process resources for the Future1 process versus the current process is that there are only two human resources participating in the Order Handling process: Order Manager and Shipper. The Customer Service Representative is not involved in this process any longer, the Web application is performing this function now.

After costs and availability for the Order Manager and Shipper are entered into the Modeler these two resource attributes look as shown in Figure 8-4.

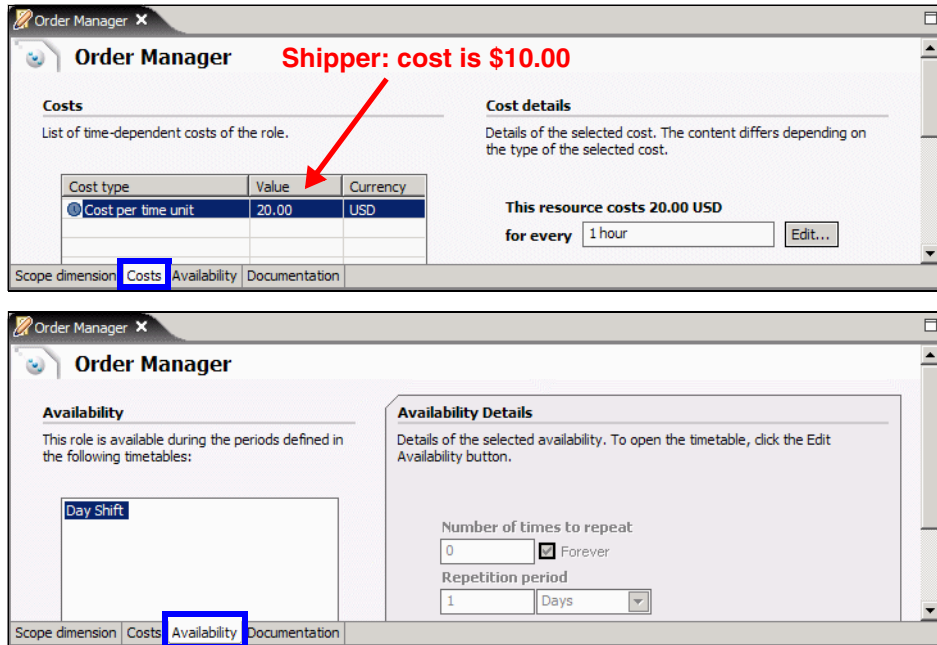


Figure 8-4 Order Manager cost and availability

Populating activity duration

The next step is to enter the processing time for the process activities. The activity duration represents the time while the activity is actively executing, rather than the elapsed time, which may include delays while waiting for a resource. It is also possible to specify the maximum amount of time that the task should wait for a resource before failing.

For our business case we will assign two types of duration information, one for the activity and one for the human tasks (role resource):

- ▶ To set the whole activity processing time, open the Order Handling Future 1 process.
- ▶ In the process flow, select the activity, select the *Duration* tab, and enter the duration value for each of the Order Handling Future 1 process activities as shown in Figure 8-5 using the data from Figure 8-3 on page 167.

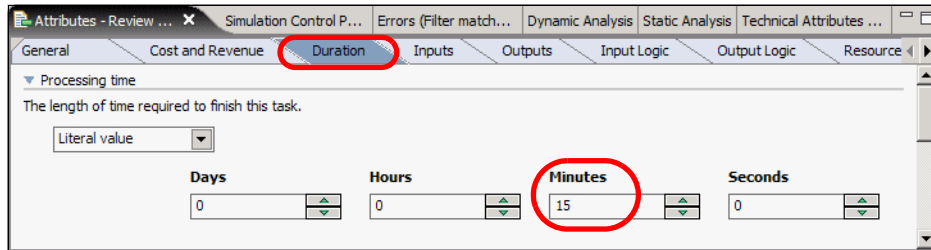


Figure 8-5 Activity duration for the Review Order activity

The activity durations populated are:

- ▶ Check Order Handling Policy: 1 second
- ▶ Check Customer Account Status: 1 second
- ▶ Review Order: 15 minutes
- ▶ Ship Order to Customer: 16 minutes
- ▶ Cancel Order and Send Notification: 1 second
- ▶ Update Order Database: 1 second
- ▶ Receive Order (in Overall Order process): 7 minutes 20 seconds

The resource durations are entered on the Resources tab under Role requirements (for people) and Bulk requirements (for systems) as shown in Figure 8-6 for the Ship Order to Customer activity.

Enter the following resource durations:

- ▶ Review Order:
 - Order Manager: 15 minutes
- ▶ Ship Order to Customer:
 - Shipper: 15 minutes
- ▶ Receive Order (in Overall Order process):
 - Customer: 7 minutes 20 seconds
 - Web application: 30 seconds

Attributes - Ship Or... x Simulation Control P... Errors (Filter match... Dynamic Analysis Static Analysis Technical Attributes ...

Cost and Revenue Duration Inputs Outputs Input Logic Output Logic Resources Organi

▼ Role requirements

This section displays the list of role requirements.

Name	Role	Time required	Quantity	Unit of meas...	Resource de...
Role requirement: 1	Shipper	15 minutes	1	units	Staff

▶ Individual resource requirements
 ▶ Bulk resource requirements

Figure 8-6 Resource durations for Ship Order to Customer

Populating probabilities on decision output branches

Enter probability to each output branch on all decisions to indicate the probability of that branch executing at any given time. Open the process flow, select every decision you plan to change, select the *Output branches* tab, and enter the probability values assembled in Figure 8-3 on page 167.

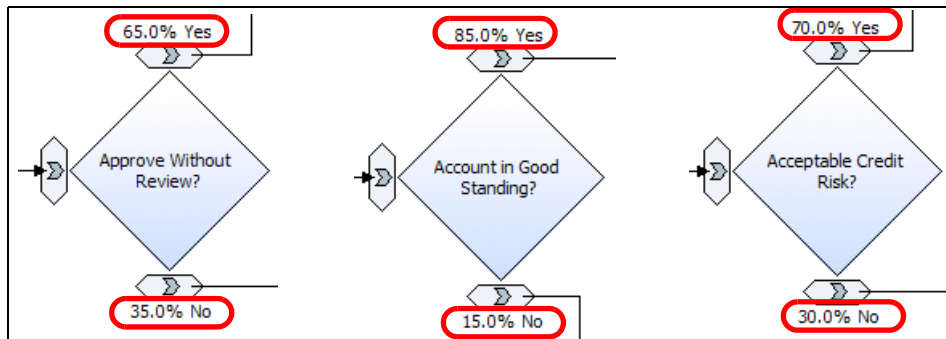


Figure 8-7 Probabilities on output branches

Once you have populated all business process values in the process model, you are ready to create a simulation snapshot and then run a simulation for the Order Handling Future 1 process.

Simulating the Future 1 process

In this section we simulate the Overall Order (Future 1) process, which calls the Order Handling (Future 1) process.

Process instance simulation

Before running the simulation on your process, you have to create a simulation snapshot for this process and add a simulation profile with the appropriate attributes.

Simulation profile information

Process simulation attributes define conditions and behavior for a process for a duration of a simulation run. We will simulate 540 tokens arriving every 2 minutes.

Note: For more information about simulation attributes and distribution models, refer to the product documentation: *Simulation processes* → *Setting Simulation Attributes* → *Specifying token creation settings*.

Creating the simulation snapshot

To create a simulation profile, select the Overall Order (Future 1) process in the Project Tree and *Simulate* in the context menu.

For detailed instructions, refer to “Creating a simulation snapshot” on page 103. Under the snapshots you will find the Overall Order (Future 1) -date- settings and the Overall Order (Future 1) -date- profile.

Populate the simulation snapshot

Open the Defaults and set the simulating attributes (Figure 8-8).

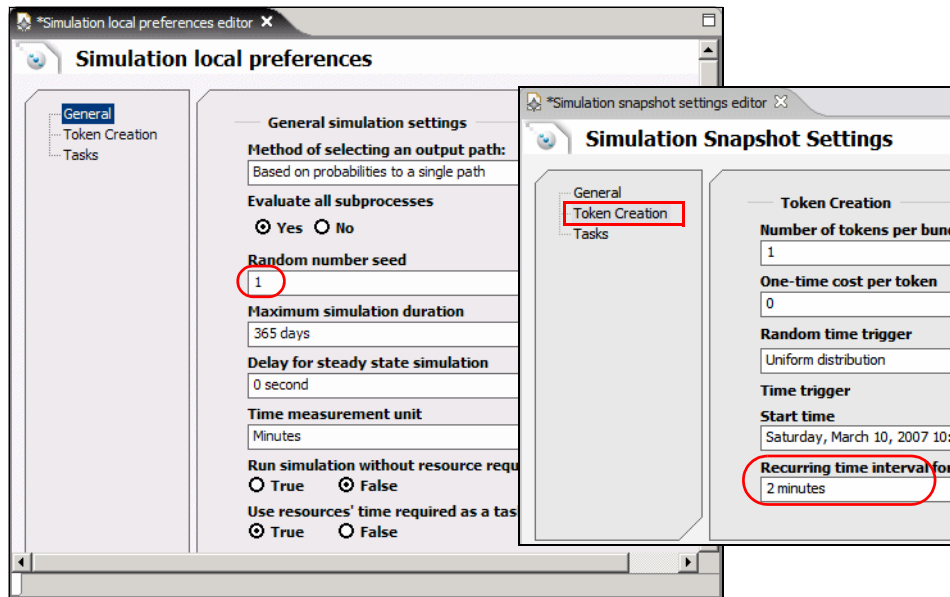


Figure 8-8 Simulation local preferences

Populate the simulation process profile

Open the profile and in the Attributes view, General tab, set these values (Figure 8-9):

- ▶ Starting date and ending date: leave the default
- ▶ Evaluate all subprocesses check: *Yes*
- ▶ Time measurement unit: *Minutes*
- ▶ Maximum simulation duration: *365 days*
- ▶ Delay of steady state simulation: *0*
- ▶ Method of selecting an output path: *Based on probabilities*
- ▶ Resources' time required: *Yes*

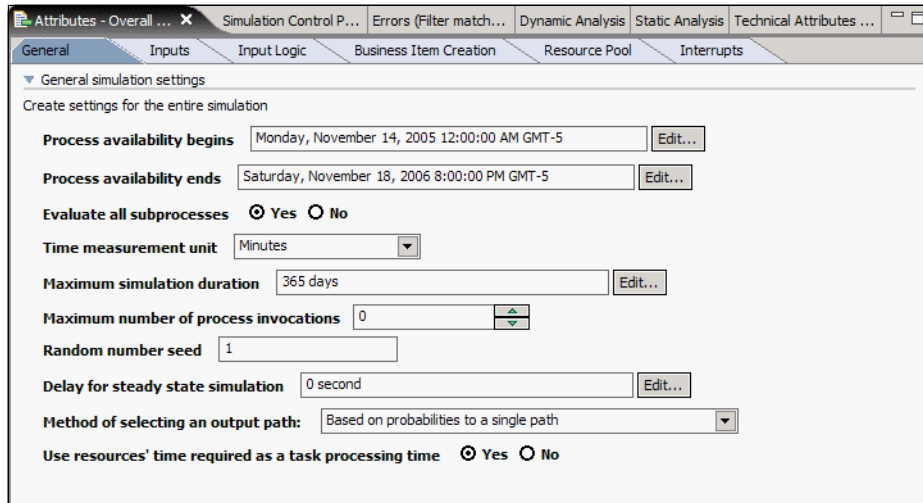


Figure 8-9 Simulation profile: General

Next, populate the *Inputs* tab as shown in Figure 8-10:

- ▶ Click on the row with *Input*.
- ▶ Total number of tokens: **540** (2 days of 270 requests per day).
- ▶ Select *Timetable trigger*.
- ▶ For the timetable, click *Browse* and select the *Online Request* timetable.
- ▶ For the recurring time interval, set the frequency to 5 minutes 20 seconds (2880 minutes for 2 days divided by 540 tokens).
- ▶ Set the maximum tokens per interval to 270.

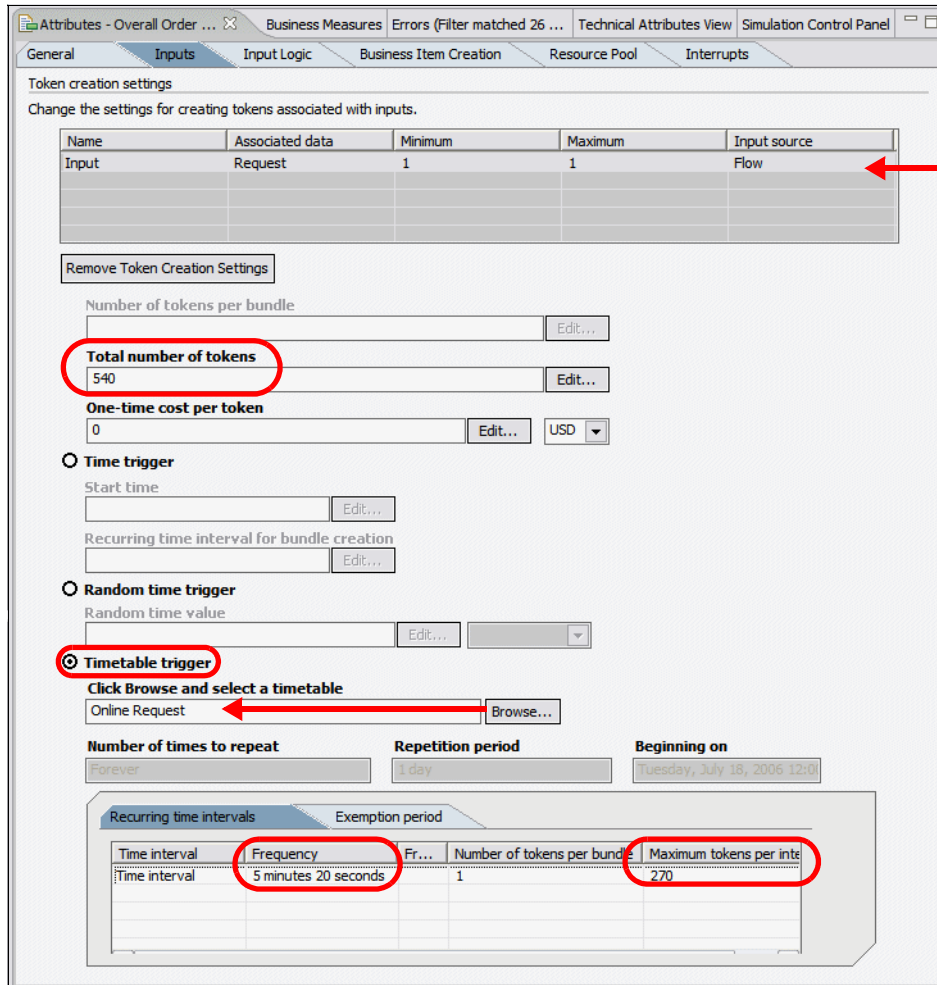


Figure 8-10 Simulation profile: Inputs

On the *Resource Pool* tab, for the three roles, deselect *Unlimited* and set the quantity to one (1), except that we employ two shippers (Figure 8-11).

Note: The Shipper was the main bottleneck in the current business process. Therefore we increase the number of shippers to two.

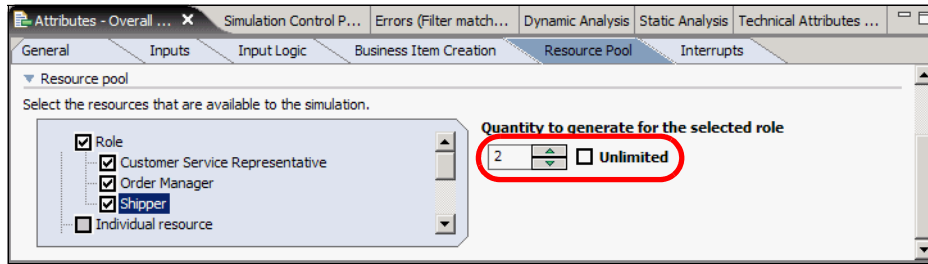


Figure 8-11 Simulation profile: Resource pool

Validate the simulation attributes


To validate the simulation attributes, select the Overall Order (Future 1) simulation profile and *Profile Analysis* → *Profile Specification* → *Select All*.

This function displays the resources with costs and durations by activity as shown in Figure 8-12.

Profile Specification Overall Order (Future 1) Saturday, March 10, 2007 12:43:32 PM PST 12:56 PM						
Activity Name	Profile Specification Overall Order (Future 1) Saturday, March 10, 2007 12:43:32 PM PST 12:56:50 PM PST					Distribu. ^
Order Handling (Future 1)						
Order Handling (Future 1)/Acceptable Credit Risk?						Yes 70.00 No 30.00
Order Handling (Future 1)/Account in Good Standi...						Yes 85.00 No 15.00
Order Handling (Future 1)/Approve Without Review?						Yes 65.00 No 35.00
Order Handling (Future 1)/Cancel Order and Send ...	1 second					
Order Handling (Future 1)/Check Customer Accou...	5 seconds					
Order Handling (Future 1)/Check Order Handling P...	1 second					
Order Handling (Future 1)/Review Order	15 minutes					
		Role	Order Manager	15 minutes		
Order Handling (Future 1)/Ship Order to Customer	16 minutes					
		Role	Shipper	15 minutes		
Order Handling (Future 1)/Update Order Database	1 second					
Receive Order	7 minutes 20 ...					
		Bulk Res...	Customer	7 minutes 20 ...		
		Bulk Res...	Web Application	30 seconds		

Figure 8-12 Profile analysis

Running the simulation

To run the simulation, use the Simulation Control Panel view (behind the Attributes view), and click the green arrow  to start the simulation (Figure 8-13).

See “Running the simulation” on page 109 for instructions on how to run with animation or without animation (use the down arrow and select *Settings*).

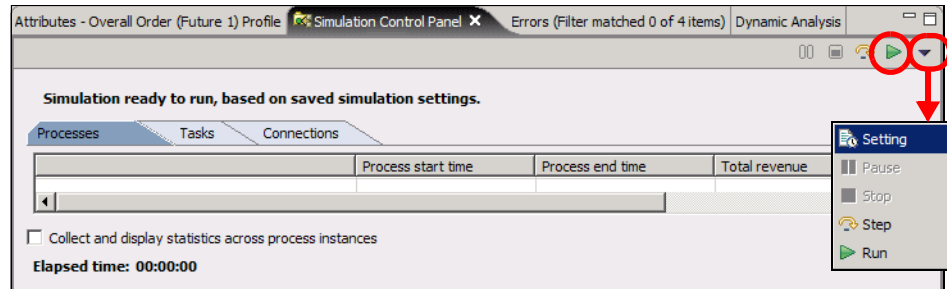


Figure 8-13 Simulation Control Panel

Analyzing the Future 1 simulation results

Once the simulation of the process is complete, the business analyst generates the same reports as for the current process. He performs an analysis based on the simulation data presented in the Modeler's dynamic analysis and reporting capabilities. He compares the revised process results with those of the current process to be sure that the new process will help meet the objectives set by management.

An initial assessment is made using the Simulation Control Panel Tasks view (Figure 8-14).

	Total instances
Cancel Order and Send Notification	68
Check Customer Account Status	347
Check Order Handling Policy for Automatic Ap...	540
Receive Order	540
Review Order	229
Ship Order to Customer	472
Update Order Database	472

Figure 8-14 Simulation Control Panel: Tasks

Of the 540 orders, 472 were shipped, which is about 87%.

The business analyst runs the following specific reports by selecting the results and *Dynamic Analysis*:

- ▶ **Process duration**—Shows average duration of each process case
- ▶ **Process cases summary**—Shows the percentage of each possible case in the process, for example, the percentage of orders shipped without requiring approval
- ▶ **Resource usage**—Shows resource usage for each process instance, including the duration of resource shortage when this resource is not available due to the work on another task
- ▶ **Process cost**—Shows average cost for each process case

The business analyst also runs process comparison reports and compares the results of the revised process with those of the current process to ensure that the revisions he has made are achieving the desired objectives.

When the business analyst and the management team are satisfied with the data that the simulations have produced, the business analyst is ready to create a business measures model that will enable the company to monitor, analyze, and report on actual runtime data.

Process duration

Process duration analysis shows the process instance elapsed duration and throughput details for each process case in a simulation.

To get the process duration information, select the simulation result element and *Dynamic Analysis* → *Process Cases Analysis* → *Process Duration* (Figure 8-15).

Process elapsed duration includes transfer times between activities and the elapsed durations of all activities on a path (called a *case*). A process case is defined as a set of process instances that have the same processing path. Calculations are performed per case by getting the simple average of all the process instance duration records in a case.

Case Name	Distribution	Success Status	Average Elapsed Duration	Average Throughput
Case 1	57.59%	Succeeded	2 days 20 hours 33 minutes 32.941 seconds	0.01 work item / hour
Case 2	24.81%	Succeeded	5 days 12 hours 11 minutes 49.939 seconds	0.01 work item / hour
Case 3	10.93%	Succeeded	3 days 13 hours 58.406 seconds	0.01 work item / hour
Case 4	5.00%	Succeeded	4 days 20 hours 18 minutes 0.184 seconds	0.01 work item / hour
Case 5	1.67%	Succeeded	4 days 5 hours 57 minutes 17.888 seconds	0.01 work item / hour
All Cases			3 days 17 hours 5 minutes 33.164 seconds	0.01 work item / hour

Figure 8-15 Analysis: Process duration

- ▶ The first case is for an order that was automatically approved without the review and **shipped** to a customer.
- ▶ The second case reflect a **shipped** product to a customer with the required review and approval by the Order Manager.
- ▶ The third case is for a **cancelled** order for a customer that was not automatically approved by the system and the Order Manager cancelled the order due to an unacceptable credit risk.
- ▶ The fourth case reflects a **shipped** product to a customer for an order that was automatically approved without the review, but the review by the Order Manager was required because the customer account was not in good standing.
- ▶ The fifth case is for a **cancelled** customer order that was approved without the Order Manager review initially, but was then cancelled by the Order Manager due to an unacceptable credit risk.

Note: The execution path of the individual cases, that is, whether the order is shipped or not, is derived from the Process Case Summary, which we complete next.

Conclusion of the business analyst

At this point, we can validate that on 540 orders, about 87.4% were shipped to customers and the average duration of the revised process is 3 days and 17 hours.

Process cases summary

The process cases summary analysis shows summary details for all the process cases produced during the simulation of a process.

To get this information, select the simulation result element and *Dynamic Analysis* → *Process Cases Analysis* → *Process Cases Summary* (Figure 8-16).

When you expand a case, you can see which activities are on the path of a case and where the most time is spent.

Process Duration Simulation result Saturday, March 10, 2007 1:02:1...				Process Cases Summary Simulation result Saturday, March 10, 20...			
Case Name	Activity...	Av...	Average Ela...	Average Resource Duration	Number of Instances	Distribution	Success Status
+ Case 1		US...	2 days 20 h...	22 minutes 50 seconds	311	57.59%	Succeeded
+ Case 2		US...	5 days 12 h...	37 minutes 50 seconds	134	24.81%	Succeeded
+ Case 3		US...	3 days 13 h...	22 minutes 50 seconds	59	10.93%	Succeeded
+ Case 4		US...	4 days 20 h...	37 minutes 50 seconds	27	5.00%	Succeeded
+ Case 5		US...	4 days 5 ho...	22 minutes 50 seconds	9	1.67%	Succeeded
All Cases		US...	3 days 17 h...	27 minutes 18.333 seconds	540	100.00%	

Process Duration Simulation result Saturday, March 10, 2007 1:02:1...				Process Cases Summary Simulation result Saturday, March 10, 20...			
Case Name	Activity Name	Average Cost	Average Elapsed Duration	Average Working Duration			
+ Case 1		USD2.51	2 days 20 hours 33 minutes 32.941 seconds	22 minutes 27 seconds			
- Case 2		USD7.51	5 days 12 hours 11 minutes 49.939 seconds	37 minutes 22 seconds			
	Order Handling (Future 1)	USD7.50	5 days 2 hours 41 minutes 41.029 seconds	30 minutes 2 seconds			
	Acceptable Credit Risk?	USD0.00	0 seconds	0 seconds			
	Approve Without Review?	USD0.00	0 seconds	0 seconds			
	Check Order Handling Policy for A...	USD0.00	1 second	1 second			
	Merge	USD0.00	0 seconds	0 seconds			
	Merge:2	USD0.00	0 seconds	0 seconds			
	Review Order	USD5.00	3 days 12 hours 41 minutes 37.611 seconds	15 minutes			
	Ship Order to Customer	USD2.50	1 day 14 hours 1.417 seconds	15 minutes			
	Update Order Database	USD0.00	1 second	1 second			
	Overall Order (Future 1)	USD7.51	5 days 12 hours 11 minutes 49.939 seconds	37 minutes 22 seconds			
	Receive Order	USD0.01	9 hours 30 minutes 8.90 seconds	7 minutes 20 seconds			
- Case 3		USD5.01	3 days 13 hours 58.406 seconds	22 minutes 22 seconds			
	Order Handling (Future 1)	USD5.00	3 days 4 hours 42 minutes 1.83 seconds	15 minutes 2 seconds			
	Acceptable Credit Risk?	USD0.00	0 seconds	0 seconds			
	Approve Without Review?	USD0.00	0 seconds	0 seconds			
	Cancel Order and Send Notification	USD0.00	1 second	1 second			
	Check Order Handling Policy for A...	USD0.00	1 second	1 second			
	Merge	USD0.00	0 seconds	0 seconds			
	Review Order	USD5.00	3 days 4 hours 41 minutes 59.83 seconds	15 minutes			
	Overall Order (Future 1)	USD5.01	3 days 13 hours 58.406 seconds	22 minutes 22 seconds			
	Receive Order	USD0.01	8 hours 18 minutes 56.57 seconds	7 minutes 20 seconds			
+ Case 4		USD7.51	4 days 20 hours 18 minutes 0.184 seconds	37 minutes 27 seconds			
+ Case 5		USD5.01	4 days 5 hours 57 minutes 17.888 seconds	22 minutes 27 seconds			
All Cases		USD4.31	3 days 17 hours 5 minutes 33.164 seconds	26 minutes 53.546 seconds			

Figure 8-16 Analysis: Process cases summary

Conclusion of the business analyst

All process cases run significantly faster, and the most time consuming activity of Review Order has to be analyzed and possibly staffed with more than one person. For example, two Order Managers can handle the outstanding backlog faster than one. This will result in improved customer satisfaction with the faster order turn around,

Resource usage

This analysis shows information about usage of each resource that is allocated in a process simulation.

To get this information, select the simulation result element and *Dynamic Analysis* → *Aggregated Analysis* → *Resource Usage* (Figure 8-17).

Resource or Role Name		A.	A	Allocating Process Instance ...	Allocating Ac...	Allocation Duration	Shortage Duration
Customer						7 minutes 20 se...	9 hours 2 minutes 18.395 seconds
Order Manager						14 minutes 44,5...	3 days 8 hours 7 minutes 16.18 seconds
	S.. S.			Overall Order (Future 1) 2	Review Order	15 minutes	0 seconds
	S.. S.			Overall Order (Future 1) 5	Review Order	15 minutes	0 seconds
	S.. S.			Overall Order (Future 1) 6	Review Order	15 minutes	7 minutes 40 seconds
	S.. S.			Overall Order (Future 1) 12	Review Order	15 minutes	0 seconds
	S.. S.			Overall Order (Future 1) 13	Review Order	15 minutes	7 minutes 45 seconds
	S.. S.			Overall Order (Future 1) 16	Review Order	12 minutes	45 seconds
	S.. S.			Overall Order (Future 1) 16	Review Order	3 minutes	15 hours
	S.. S.			Overall Order (Future 1) 18	Review Order	15 minutes	15 hours 1 minute 5 seconds
	S.. S.			Overall Order (Future 1) 20	Review Order	15 minutes	15 hours 1 minute 25 seconds
	S.. S.			Overall Order (Future 1) 22	Review Order	15 minutes	15 hours 1 minute 45 seconds
	S.. S.			Overall Order (Future 1) 503	Review Order	15 minutes	5 days 11 hours 11 minutes 25 seconds
	S.. S.			Overall Order (Future 1) 505	Review Order	15 minutes	5 days 11 hours 11 minutes 45 seconds
	S.. S.			Overall Order (Future 1) 506	Review Order	15 minutes	5 days 11 hours 19 minutes 25 seconds
	S.. S.			Overall Order (Future 1) 509	Review Order	15 minutes	5 days 11 hours 12 minutes 25 seconds
	S.. S.			Overall Order (Future 1) 516	Review Order	15 minutes	5 days 10 hours 36 minutes 5 seconds
	S.. S.			Overall Order (Future 1) 517	Review Order	15 minutes	5 days 10 hours 43 minutes 45 seconds
	M.. M			Overall Order (Future 1) 522	Review Order	15 minutes	6 days 1 hour 22 minutes 5 seconds
	M.. M			Overall Order (Future 1) 529	Review Order	15 minutes	6 days 45 minutes 40 seconds
	M.. M			Overall Order (Future 1) 532	Review Order	15 minutes	6 days 38 minutes 45 seconds
	M.. M			Overall Order (Future 1) 535	Review Order	15 minutes	6 days 31 minutes 45 seconds
	M.. M			Overall Order (Future 1) 536	Review Order	15 minutes	6 days 39 minutes 25 seconds
	M.. M			Overall Order (Future 1) 538	Review Order	15 minutes	6 days 39 minutes 45 seconds
	M.. M			Overall Order (Future 1) 539	Review Order	15 minutes	6 days 47 minutes 25 seconds
Shipper						13 minutes 54.5...	2 days 1 hour 21 minutes 12.357 sec...
Web Application						30 seconds	9 hours 2 minutes 18.395 seconds

Figure 8-17 Analysis: Resources

Conclusion of the business analyst

The human tasks performed by the Order Manager are now the bottleneck in the system. ClipsAndTacks may decide to monitor this process further to see if these shortage situations are affecting customer satisfaction. The new Web application, which replaced the Customer Service Representative and is available 24 x 7, does resolve the resource shortage duration issues of the current Receive Order process.

Process cost

The process cost analysis shows the average and weighted average costs and revenue for all process instances in each case of the simulation result.

To get this information, select the simulation result element and *Dynamic Analysis* → *Process Cases Analysis* → *Process Cost* (Figure 8-18).

Process Duration Simulation result Saturday, March 10, 2007 1:02:14 ...				Process Cost Simulation result Saturday, March 10, 2007 1:02:14 PM			
Case Name	Distribution	Success Status	Ave...	Average Delay Cost	Average Resource Cost	Average Cost	Average Profit
Case 1	57.59%	Succeeded	USD...	USD0.00	USD2.51	USD2.51	(USD2.51)
Case 2	24.81%	Succeeded	USD...	USD0.00	USD7.51	USD7.51	(USD7.51)
Case 3	10.93%	Succeeded	USD...	USD0.00	USD5.01	USD5.01	(USD5.01)
Case 4	5.00%	Succeeded	USD...	USD0.00	USD7.51	USD7.51	(USD7.51)
Case 5	1.67%	Succeeded	USD...	USD0.00	USD5.01	USD5.01	(USD5.01)
All Cases			USD...	USD0.00	USD4.31	USD4.31	(USD4.31)

Figure 8-18 Analysis: Process cost

Conclusion of the business analyst

We see that the Order Handling Future 1 process costs have come down to \$4.31 from the \$11.57 for the current process. The average costs for the Future 1 five case processes are ranging from \$2.51 to \$7.51.

Process comparison analysis

You can perform several kinds of analysis to compare the weighted average analysis results for two simulated processes that use the same input parameters.

To perform a comparison analysis, select one of the simulation results that you want to analyze and *Dynamic Analysis* → *Processes Comparison Analysis* and then one of the following choices:

- ▶ Processes Duration Comparison
- ▶ Processes Activities Total Time Comparison
- ▶ Processes Cost Comparison
- ▶ Processes NPV / IRR Comparison
- ▶ Processes Break Even Comparison
- ▶ Processes Resources Time Comparison
- ▶ Processes Resources Cost Comparison
- ▶ Processes Classifier Duration Comparison
- ▶ Processes Classifier Cost Comparison

A dialog opens where you select the second simulation results that you want to compare with the first results. Click *OK*.

Select the type of process instances you want to include in the analysis:

- ▶ Succeeded process instances only
- ▶ Failed process instances only
- ▶ All process instances

Note: You must have simulation results for the current and the future process in the same project to be able to compare results.

If your simulations are in different projects:

- ▶ Export both projects with simulation snapshots.
- ▶ Create a test project.
- ▶ Import both exported projects into the test project.
- ▶ Rerun the simulations.
- ▶ Run the comparison report.

Processes duration comparison

This analysis compares the average elapsed duration results for two process simulations.

For each process considered in the comparison, this analysis displays the information shown in Figure 8-19.

Simulation Result ...	Process Name	Average Elapsed Duration	Average Throughput
Simulation result ...	Overall Order (Future 1)	3 days 17 hours 5 minutes 33.164 seconds	0.01 work item / h...
Simulation result ...	Order Handling (Current)	10 days 8 hours 13 minutes 43.944 seconds	0.00 work item / h...
Difference		-6 days 15 hours 8 minutes 10.78 seconds	0.01 work item / h...
Percentage Change		-178.62%	64.11%

Figure 8-19 Comparison: Process duration

Calculated values are based on weighted average values calculated according to the process duration analysis. The following information is displayed in the process duration comparison:

- ▶ Simulation Result Name—The name and the time stamp of the simulation result to which the process belongs.
- ▶ Process Name—The name of the process.
- ▶ Average Elapsed Duration—Average elapsed duration of the process SUM (case Average Elapsed Duration * case Distribution / cases total Distributions) for all cases.
- ▶ Average Throughput—Average throughput of the process SUM (case Average Throughput * case Distribution / cases total Distributions) for all cases.

For each column that displays a numerical result, the following comparative information is also displayed:

- ▶ Difference—Calculated as: first process value - second process value
- ▶ Percentage change—Calculated as: (difference / first process value) * 100

This reports shows a large improvement in the process duration metrics for the Order Handling Future 1 process.

Processes cost comparison analysis

This analysis compares the average cost and revenue results for two process simulations that use the same input parameters.

For each process considered in the comparison, this analysis displays the information shown in Figure 8-20. Calculated values are based on weighted average values calculated according to process cost analysis.

For each column that displays a numerical result, the following comparative information is also displayed:

- ▶ Difference—Calculated as: first process value - second process value
- ▶ Percentage change—Calculated as: (Difference / first process value) * 100

	Simulation Result...	Process Name	Average Resource Cost	Average Cost	Average Profit
	Simulation result...	Overall Order (Future 1)	USD4.31	USD4.31	(USD4.31)
	Simulation result...	Order Handling (Current)	USD11.57	USD11.57	(USD11.57)
Difference			(USD7.25)	(USD7.25)	USD7.25
Percentage Change			-168.12%	-168.12%	-168.12%

Figure 8-20 Comparison: Process cost

This report shows a 170% improvement in the process cost metrics for the Order Handling Future 1 process.

More information

For more information about comparison analysis, refer to the product documentation under *Analyzing models and simulations*.

Additional considerations

There has been a substantial drop in price, but a seasoned business analyst would look at the costs outside of those generated by the Modeler, or add additional costs to the model based on discussions with the systems architect, who will build the application.

Specifically, the business analyst would consider the costs for buying, licensing, developing, and maintaining the proposed application. Also, the business analyst would check the assumptions about how long the customer might spend entering the order information using the Web application with a usability expert.

Some questions that the systems architect would ask the business analyst:

- ▶ How many customers will be using the application right away? How many in the near future?
- ▶ How many customers might use the application at the same time?
- ▶ Is it critical to guarantee completion of an order (in which case a messaging system would have to be added to the expense) or is an occasional failure in the system with a returned error message acceptable?
- ▶ Should the application have a service department that a customer could reach at any time?

The answers to these questions would drive the total cost of the application.

Perspective

A wise business analyst would put the analysis figures in perspective;

- ▶ For example, even with the dramatic improvement promised by the online application, if ClipsAndTacks were a small corporation and the costs for creating the application were substantial, then the business analyst would still not proceed.
- ▶ However, if the ClipsAndTacks were a medium to large size corporation with in-house developers and most of the software already purchased, then the online application would make more sense—not just from an efficiency point of view, but also from the opportunity to increase sales by offering the service nationally (or internationally) on the Web.

Summary

In this chapter we simulated the Order Handling (Future 1) process and analyzed the results. We can see a major improvement by having the customer handle the data entry for orders. The implementation of the automatic approval and the customer credit check also helps for the overall achievement of 90% shipped orders.

The model with simulation snapshot is exported as:

```
SG247148\sampcode\model\Clips And Tacks Future 1 with Simulation.mar
```




Defining KPIs and measures

This chapter describes the steps performed in WebSphere Business Modeler Version 6.0.2, for creating business measures. We define all the relevant information about the process that will be monitored by WebSphere Business Monitor Version 6.0.2.

In Chapter 8, “Simulating and analyzing the Future 1 process” on page 163, we worked with the simulation of the process. Now we have the opportunity to measure the same process in action, enabling you to measure the performance of the process, retrieving the information that flows through the Order Handling (Future 1) process.

The following topics are outlined in this chapter:

- ▶ Business measures
- ▶ Defining ClipsAndTacks key performance indicators (KPIs)
- ▶ Defining measures
- ▶ Dimensional analysis
- ▶ Exporting the business measures model

Business measures

In this section the we discuss the following topics:

- ▶ Introduction to business measures
- ▶ Model, assemble, deploy and manage the complete lifecycle
- ▶ Elements of the Business Measures editor
- ▶ WebSphere Business Monitor views

Introduction to business measures

A business measures model describes business metrics, their dependencies on incoming events, conditions warranting business action (business situations), and outbound events that represent notifications of such conditions and might trigger business actions.

Specifically, the business measures model describes how to perform the following actions:

- ▶ Gather information, from real-time events.
- ▶ Aggregate information to calculate higher-level business metrics and key performance indicators (KPIs).
- ▶ Represent the calculated values on a number of dashboard views and reports, based on the business needs.
- ▶ Emit events in reaction to business conditions that could be used to trigger actions.

The key to having a successful set of business measures is deciding upon the vital measures that are linked to your business success. You study the process and the business goals to determine which business measures will be needed from the executing process. Once the measures have been established, you can monitor and evaluate them.

Model, assemble, deploy, and manage the complete life cycle

WebSphere Business Modeler enables you to transform business process models to IT-level models. You can export Business Process Execution Language (BPEL) and business measures, then use WebSphere Integration Developer to import the BPEL model and assemble the services and components. Once you finish the assembly, you can deploy the application to WebSphere Process Server for testing.

You can import the business measures into WebSphere Business Monitor Toolkit in Integration Developer and refine these business measures into a monitoring application by attaching them to business process events. You can deploy the monitoring application to a Monitor Server test environment to monitor and evaluate the business process at runtime. The finished business measures monitoring application can be deployed to WebSphere Business Monitor Server to monitor the real application running in Process Server (Figure 9-1).

Note: In Version 6.0.2 you can also use the Business Monitor Server to run the business process application (simplified environment).

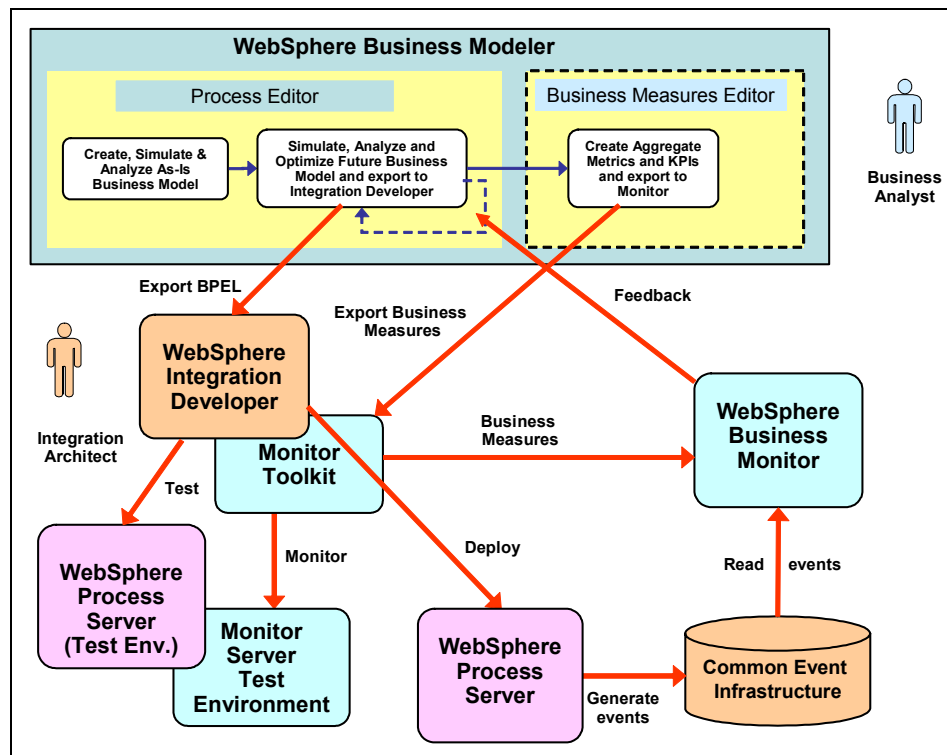


Figure 9-1 BPM complete life cycle

WebSphere Business Monitor displays dashboards (Figure 9-2), which are containers (portals) that enable you to monitor different aspects of business performance. You can use WebSphere Business Monitor to capture near real-time, work-in-process items and perform corrective actions by reassigning, re-prioritizing, or suspending them. You can also display data from work items produced by the runtime, and retrieve and view the historical data of the monitored process.

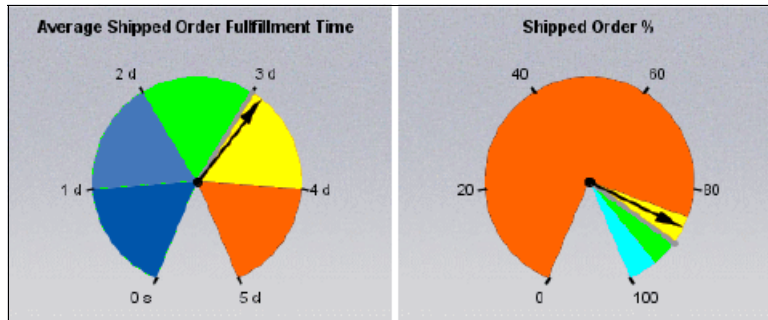


Figure 9-2 Dashboard example

Elements of the Business Measures editor

You can combine WebSphere Business Modeler with WebSphere Business Monitor for business performance management, the process of continuously defining, analyzing, and improving a business process. Starting with a process model in the Modeler, you create a business measures model that enables you to specify the performance management aspects, including business metrics, key performance indicators (KPIs), and situation events.

Figure 9-3 shows the business measures elements relationship.

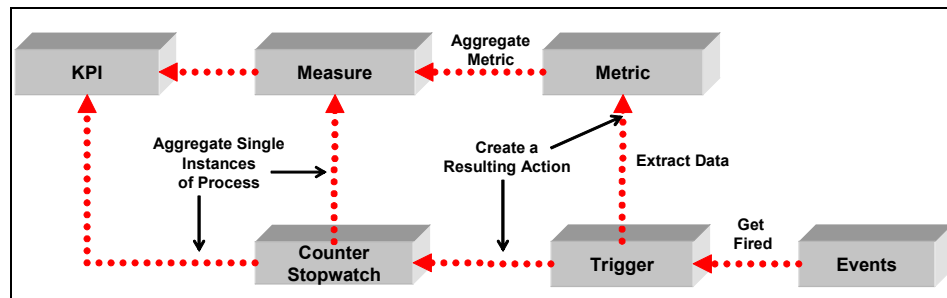


Figure 9-3 Business measures relationship

We describe the following business measures elements:

- ▶ KPIs (key performance indicators)
- ▶ Metrics and measures
- ▶ Dimensions

Key performance indicators

Key performance indicators (KPIs) are the detailed specifications required to track the performance against business objectives. Each KPI is associated with a specific process, and is quantifiable, measurable, and results-oriented. They are defined within the context of the Business Measures editor and evaluated by WebSphere Business Monitor.

KPIs are based on business objectives. KPIs enables the organizations to measure an aspect of the process against a defined target, so KPIs can be compared with actual results to determine the level of success.

A KPI can be defined on the basis of:

- ▶ Target value—Exact value of the business measure.
- ▶ Time period—Time duration of execution of a business process. Time periods can be repeating, rolling or fixed.
 - Repeating: Time period to calculate the KPI on data from a repeating time period of a specific length, for example, time period for previous day, last month, last year.
 - Rolling: Time period to evaluate the KPI over a period of days that moves continuously, for example, time period for the last 30 days. This can be the last full period or the period in progress.
 - Fixed: Time period to have the KPI calculated across a single time period, for example, time period between March 31, 2007 and March 31, 2008.
- ▶ Range—Range against which to track the KPI. Range can be either a percentage of the target value or an actual value.

Metrics

A metric is a measurement of a process or a process element that is used to assess business performance. A metric can be used alone or in combination with other metrics to define the calculation for a KPI, which measures performance against a business objective. The values that metrics return are captured and evaluated using WebSphere Business Monitor.

There are two ways to define metrics:

- ▶ **Instance metric**—Return the result from one process instance.
- ▶ **Aggregate metric**—Provide calculations across multiple runs of the process, for finding the average, count, maximum, minimum, or total number of occurrences. These aggregate metrics are also called **measures**.

Dimensions

Dimensions are data categories that are used to organize and select instances for reporting and analysis. Dimensions enable the display of numeric measures as a function of other data, for example, display the number of products ordered at a particular location, or display the best sales person for a particular product group.

Monitor Development Toolkit

The business measures model defined in the Modeler will be completed as a monitor model using the Monitor Model Editor in the Monitor Development Toolkit. We cover this task in Chapter 12, “Developing and testing the business measures with the Monitor Toolkit” on page 327.

WebSphere Business Monitor dashboard views

The key performance indicators (KPIs) and other metrics that you define in the Business Measures editor will be displayed in the WebSphere Business Monitor dashboard views, based on WebSphere Portal Server. A dashboard view is a portal page supported by one or more dashboard portlets. The display is targeted to address specific functions or responsibilities of a particular role. Each dashboard view supports a number of display properties to customize the appearance. The dashboard has the following portlets:

- ▶ **Instances:** Displays the instances of a specific process and the runtime values of selected business measures.
- ▶ **KPIs:** Displays the details of individual KPIs.
- ▶ **Gauges:** Visually represents the values of KPIs relative to KPI ranges or relative to the KPI target, in the form of a gauge that looks like an automobile speedometer or tachometer.
- ▶ **Dimensional:** Dimensional analysis provides business insight by summarizing business metrics. It organizes data into levels of detail that you can drill down to extract significant information.
- ▶ **Reports:** Displays performance reports showing the values of metrics aggregated over a period of time in tables and graphs.
- ▶ **Alerts:** Displays alerts that notify users of defined business situations occurring at runtime.
- ▶ **Diagram:** Displays the diagrams associated with particular business process.
- ▶ **Organization:** Displays the structure of the organization.
- ▶ **Export values:** Export activity processing times and decision percentages for import into WebSphere Business Modeler.

More information

The detailed documentation of business metrics and how to create business measures with WebSphere Modeler Version 6 is available in the help documentation included with the product, or you can refer to:

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

ClipsAndTacks key performance indicators

In this section the we discuss the following topics:

- ▶ Definition of ClipsAndTacks KPIs and measures
- ▶ How to create the business measures model
- ▶ Detailed steps to create two KPIs:
 - Average order fulfillment is three days or less.
 - Number of orders shipped, with a target of 90%.

Definition of ClipsAndTacks KPIs and measures

The ClipsAndTacks management team wants to be able to measure the results of the revised process when it is implemented to ensure that it is helping to meet the company's business objectives. To measure the revised order handling process, management has identified business measure for the new process.

KPIs

- ▶ Average Process Duration (only for shipped orders) with a target of 3 days or less, and a range between 1 and 4 days
- ▶ Percentage of Orders Shipped, with a target of 90% and a range between 85% and 100%

Measures

- ▶ Order Count, Shipped Order Count, and Declined Order Count
- ▶ Order Price Total and Order Price Average

Dimension

- ▶ Location: Country (level1) and city (level2) from the customer data

Alerts

- ▶ Send an alert when an instance of the process duration exceeds 3 days and 1 hour.
- ▶ Send an alert when the Percentage of Orders Shipped KPI is less then 85%.

Creating the business measures model

We work with the Order Handling (Future 1) process to define the business measures. Set the modeling mode to advanced, by selecting *Modeling* → *Mode* → *Advanced*.

Open the Order handling (Future 1) process.

Note: You can import the model populated with business measures from:

SG247148\sampcode\model\Clips And Tacks Future 1 with Measures.mar

See “Importing the current process model using the Modeler” on page 83 for instructions on how to import a model.

Preparation

The first step to create the KPIs is to show the Business Measures view that enables you to create the business measure elements:

- ▶ If you have not already done so, make the Business Measures view visible by selecting *Windows* → *Show View* → *Business Measures* (Figure 9-4).

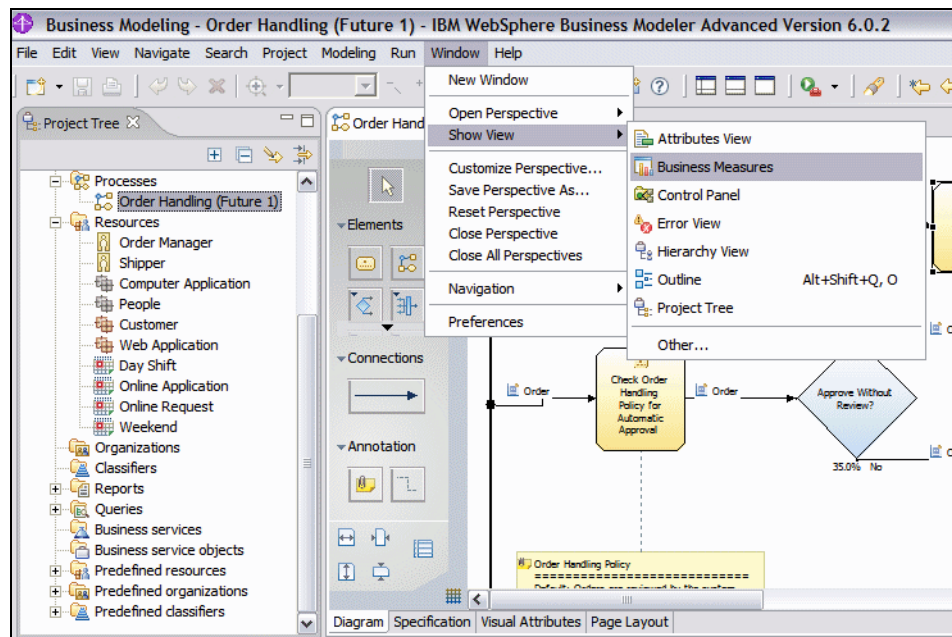


Figure 9-4 Select Business Measures

- ▶ If Business Measures are not visible, navigate to *Window* → *Show View* → *Others*, and in the Show view panel, expand *Business modeler views* → *Business Measures* (Figure 9-5).

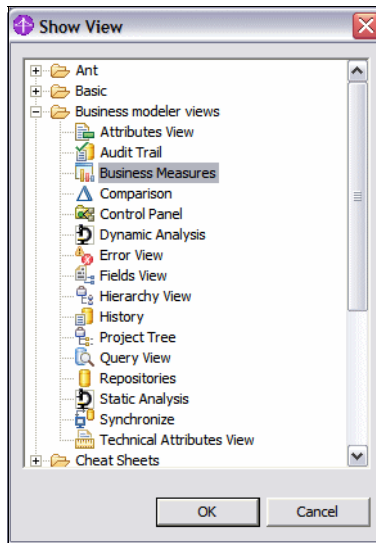


Figure 9-5 Show View for Business Measures

- ▶ Click *OK* and the Business Measures view is shown (Figure 9-5).

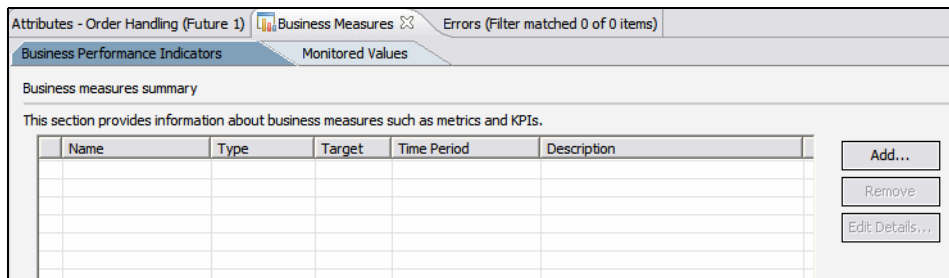


Figure 9-6 Business Measures view

- ▶ The Business Measures editor opens on the Business Performance Indicators tab.

Note: In WebSphere Business Modeler V6.0.2, there are Business Performance Indicators and Monitored Values tabs for defining KPIs and metrics.

- ▶ The Business Measures view is linked to the process diagram that is open (Figure 9-7).

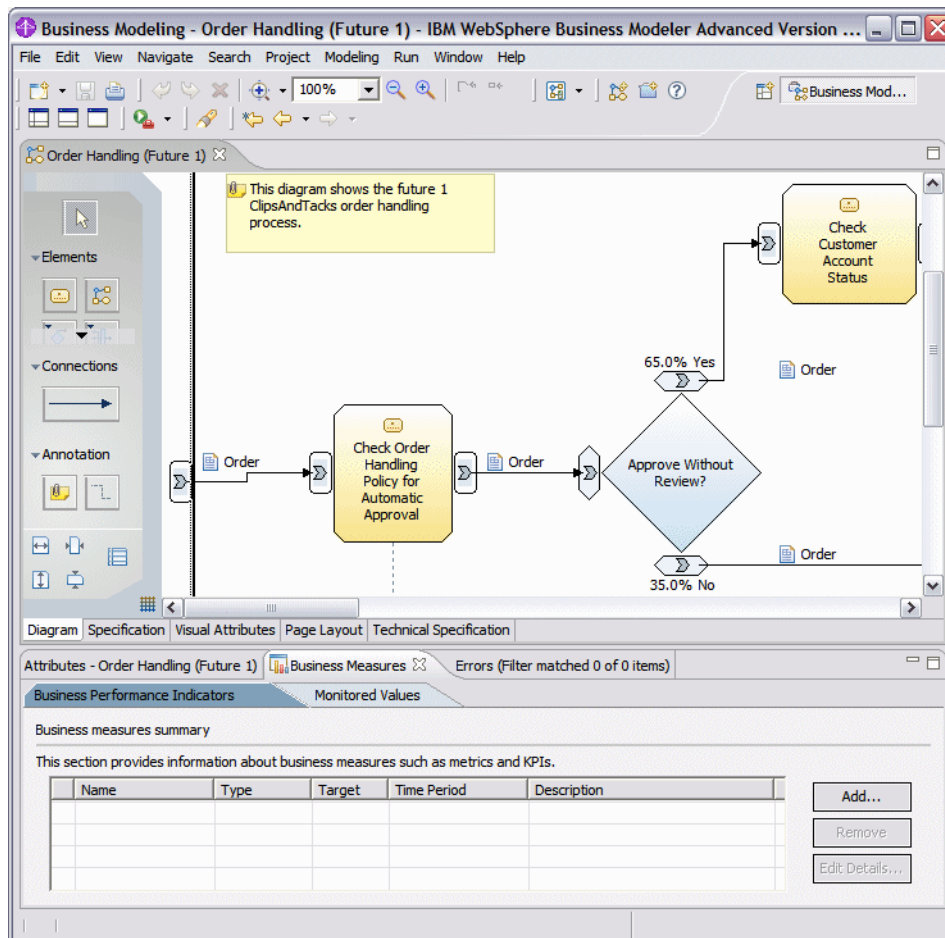


Figure 9-7 Business measures: Diagram and Business Measures view

Monitored values

Monitored values indicate which values you want returned from the Business Monitor after the process has been monitored.

- ▶ In the Business Measures view, select the *Monitored Values* tab.
- ▶ Select activities as shown in Figure 9-8.

We want to have accurate values for the activity processing time and for the decision percentages. We did not model cost and revenue in our process.

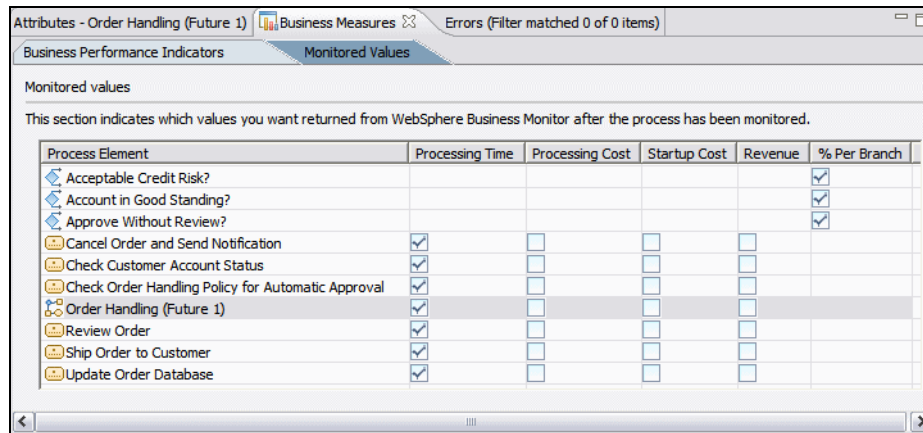


Figure 9-8 Monitored values

Defining the KPIs

Now we implement the ClipsAndTacks KPIs.

Average Process Duration

In this section we provide the detailed steps to define the Average Process Duration for Shipped Orders KPI.

The Average Process Duration KPI allows the management team to monitor (through the WebSphere Monitor) the average amount of time to ship orders to customers. Analysis of this information will help ClipsAndTacks to reduce the average time from when orders are received, to the time they are shipped, to 3 days. The simulation of the Order Handling (Future 1) process in Chapter 8, “Simulating and analyzing the Future 1 process” on page 163 indicates an average time of over 3 days to complete the execution of the process.

To define the KPI, follow these steps:

- ▶ Select the process in the diagram and the Business Performance Indicators tab in the Business Measures view.
- ▶ Click *Add* in the Business Measures view (Figure 9-7 on page 196). Specify the following details (Figure 9-9):
 - Name: Average Process Duration
 - Type: select *KPI*

- Description: Measure average time of business process duration for shipped orders.
- Select *Specify a target value and type* and select *Duration* for the Type and a target value of 3 days.

Business Measure Details - Average Process Duration

Business Measure Details
Specify additional details to describe how the business measure is calculated.

Name Average Process Duration

Type KPI Instance metric Aggregate metric Unspecified

Description Measure average time of business process duration for shipped orders.

▶ Dashboard views


Specify a target value and type
The target is an exact value that the KPI should achieve.

Type: Duration

Target value: Days: 3 Hours: 0 Minutes: 0 Seconds: 0 Milliseconds: 0

OK Cancel

Figure 9-9 Create KPI

- ▶ Scroll down and select *Specify range details* (Figure 9-10):
 - Select *Actual Value*.
 - Click *Add* twice.
 - Range names: Duration acceptable, Duration too long.
 - Start and end values: Click the  icon and select the number of days.

Note: When we implement the KPI, we can define more ranges to get a better graphical display.

Specify range details:

Ranges can be defined as percentages of the target value or as fixed, actual values.

Percentage of target value (target value = 100%)

Actual value

Specify ranges

A range is a set of values, such as allowable margins or lower and upper limits, against which to track your KPI.

Range name	Start value	End value
Duration acceptable	1 days	< 3 days
Duration too long	3 days	< 5 days

Figure 9-10 Specify range details

- Scroll down and select *Specify when to send an alert and the action to take as a result*. Click *Add* and an Average Process Duration Alert is added. Overtyping the generated alert description with: Send an alert when the process duration is longer than 3 days and 1 hour (Figure 9-11).

Specify when to send an alert and the action to take as a result

For example, when this measure exceeds a certain value, an email may need to be sent.

Alert description

Send an alert when the process duration is longer than 3 days and 1 hour.

Figure 9-11 Specify an alert

- Scroll down and select *Select the dimensions across which to calculate the KPI value*. Click *Add* and overtype the dimension name with *Location*. Then select the *Location* dimension (Figure 9-12).

Select the dimensions across which to calculate the KPI value

For example, Total profit by City, or Average salary by Date.

Dimension

Location

Figure 9-12 Specify a dimension

- Click *OK* and the KPI is created (Figure 9-13).

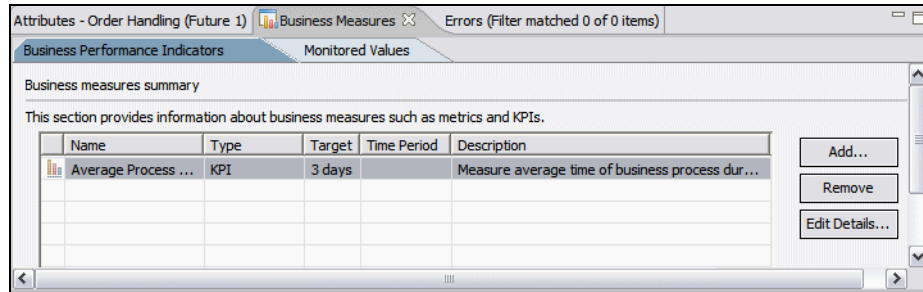


Figure 9-13 KPI is added

Percentage of Orders Shipped

In this section we provide the detailed steps to define the Percentage of Orders Shipped KPI.

The Percentage of Orders Shipped KPI allows the management team to monitor the percentage of shipped orders. Analysis of this information will help ClipsAndTacks to achieve shipping 90% of the orders (or better). The simulation of the Order Handling (Future 1) process indicates a percentage of shipped orders of close to 90%.

In the Business Measures View click *Add* and specify the details (Figure 9-14):

- ▶ Name: Percentage of Orders Shipped
- ▶ Type: select *KPI*
- ▶ Description: Percentage of orders that are shipped.
- ▶ Select *Specify a target value and type* (Type: *Number*, Target Value: 90).
- ▶ Select *Specify a time period over which the business measure will be monitored* (select *Rolling*, number of previous days: 30).
- ▶ Select *Specify Range Details*:
 - Select *Percentage of target value (target value = 100%)*. Click *Add* three times and specify the ranges:
 - Shipped orders percentage too low, 0, < 85
 - Shipped orders percentage good, 85, < 90
 - Shipped orders percentage great, 90, 100
- ▶ Select *Specify when to send an alert and the action to take as a result* and click *Add*, which creates an alert (then overwrite the description).
- ▶ Select *Select the dimensions across which to calculate the KPI value*, and select the *Location* dimension.

Business Measure Details - Percentage of Orders Shipped

Business Measure Details
Specify additional details to describe how the business measure is calculated.

Name Percentage of Orders Shipped

Type KPI Instance metric Aggregate metric Unspecified

Description Percentage of orders that are shipped

Specify a target value and type
The target is an exact value that the KPI should achieve.
Type: Number
Target value: 90

Specify a time period over which the business measure will be monitored

Repeating Rolling Fixed

Period type: Yearly

Size of period: Number of previous days: 30

Start date:

End date:

Base period on:
 Last full period
 Period in progress

Base last day on:
 Previous day
 Day in progress

Specify range details:
Ranges can be defined as percentages of the target value or as fixed, actual values.
 Percentage of target value (target value = 100%)
 Actual value

Specify ranges
A range is a set of values, such as allowable margins or lower and upper limits, against which to track your KPI.

Range name	Start value	End value
Shipped orders percentage too low	0 %	< 85 %
Shipped orders percentage good	85 %	< 90 %
Shipped orders percentage great	90 %	100 %

Specify when to send an alert and the action to take as a result
For example, when this measure exceeds a certain value, an email may need to be sent.

Alert description
Send an alert when less than 85% of orders are shipped.

Select the dimensions across which to calculate the KPI value
For example, Total profit by City, or Average salary by Date.

Dimension
 Location

Figure 9-14 KPI for percentage of shipped orders

- ▶ Click *OK* and the KPI is created (Figure 9-15).

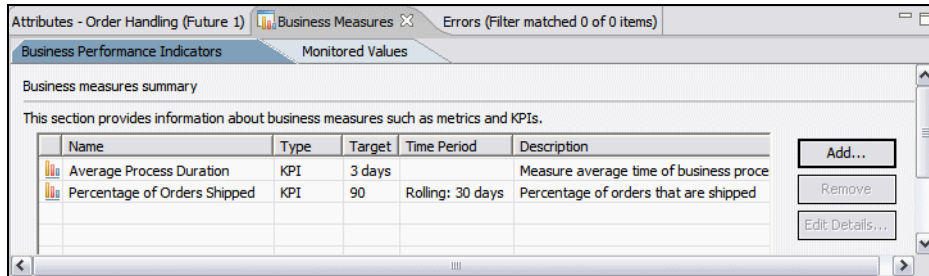


Figure 9-15 Percentage of Orders Shipped KPI Created

Define the measures

A measure is an aggregate metric (count, sum, average, minimum, maximum), based on an instance metric, which is extracted from a business process instance.

In this section we define the following measures:

- ▶ Order Count
- ▶ Shipped Order Count
- ▶ Declined Order Count
- ▶ Order Price Total
- ▶ Order Price Average

Creating the Order Count measure

We define an Order Count as a measure that counts the total number of orders.

In the Business Measures view, click *Add* and specify these values (Figure 9-16):

- ▶ Name: Order Count
- ▶ Type: *Aggregate metric*.
- ▶ Description: Count the number of orders processed.
- ▶ Select *Specify how this measure is aggregated across multiple runs of the process* and select *Count* for the function.
- ▶ Select *Specify the aspects that will be available in the Dimensional view for analysis of the metric* (the Location dimension is already defined).

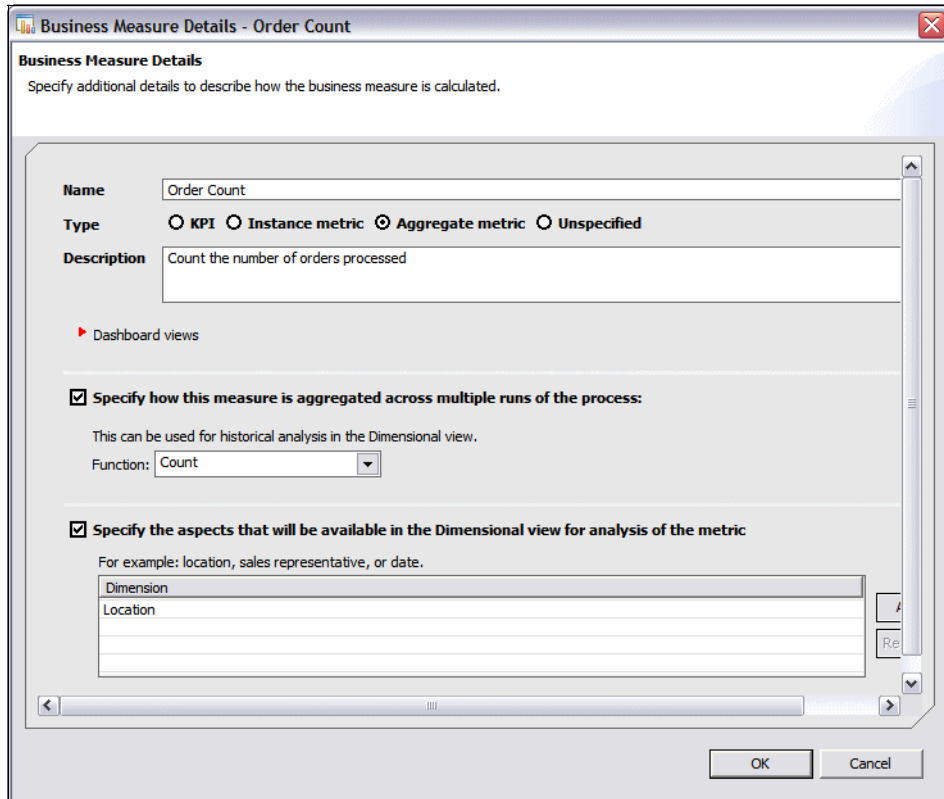


Figure 9-16 Measure Order Count

Click *OK* and the measure appears in the list (Figure 9-17).

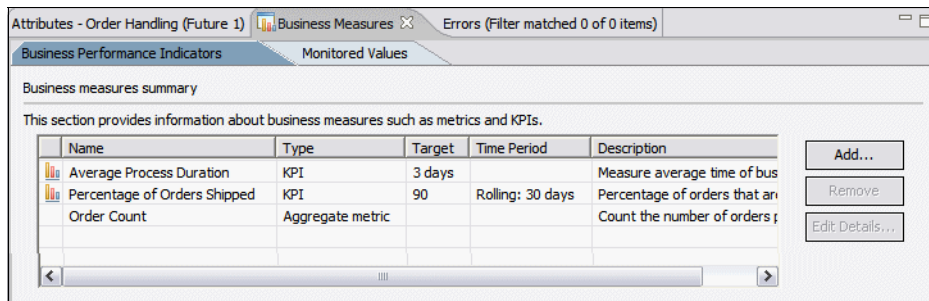


Figure 9-17 Measure Order Count created

Creating the other measures

In the same way as the Order Count measure, we define the other four measures as defined in Table 9-1. Be sure to select *Specify the aspects that will be available in the Dimensional view for analysis of the metric*.

Table 9-1 Metrics

Measure name	Description	Function
Order Count	Count the number of orders processed	Count
Shipped Order Count	Count the number of orders shipped	Count
Declined Order Count	Count the number of orders declined	Count
Order Price Total	Total value of the orders	Sum
Order Price Average	Average value of the orders	Average

The final set of KPIs and measures is shown in Figure 9-18.

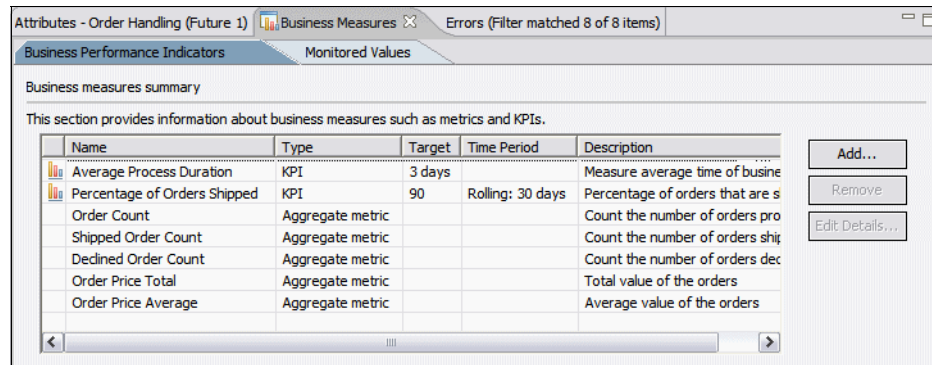


Figure 9-18 KPIs and measures for ClipsAndTacks Future 1

Dimensional analysis

Dimensional analysis organizes historical data into levels of detail that you can drill down to extract significant information.

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. Generally, any metric with non-numeric values is a level of a dimension, and you analyze numeric measures against dimensions.

For example, using dimensional analysis we can answer question like:

- ▶ Where are orders coming from? (Canada or USA)
- ▶ Are more orders declined from the USA than from Canada?
- ▶ Are we processing orders faster from cities in the USA?
- ▶ Are the orders from the USA for a higher total order price?

Basically we can analyze numerical data (counters, durations, prices) against other data that we capture (country, city).

Location dimension

For dimensional analysis we defined the Location dimension. We will define the details of this dimension in WebSphere Integration Developer. The location dimension will be composed of the country and city attributes of the customer that places an order.

Where do we define the details?

How the KPIs and measures are calculated using events, triggers, counters, and stopwatches is not defined in the Modeler. These very technical details are defined by a developer using the Monitor Development Toolkit, which is a plug-in in Integration Developer.

Refer to Chapter 12, “Developing and testing the business measures with the Monitor Toolkit” on page 327 for further information on measuring the business process.

Exporting the business measures

We export the business measures for usage in the Monitor Toolkit in WebSphere Integration Developer. Compare this export with “Exporting the model for Integration Developer” on page 157.

To export the business measures for the Monitor Toolkit, select either the project or the process and *Export* (context):

- ▶ Select the type of export you want. Select *WebSphere Business Monitor Development Toolkit* (Figure 9-19). Click *Next*.

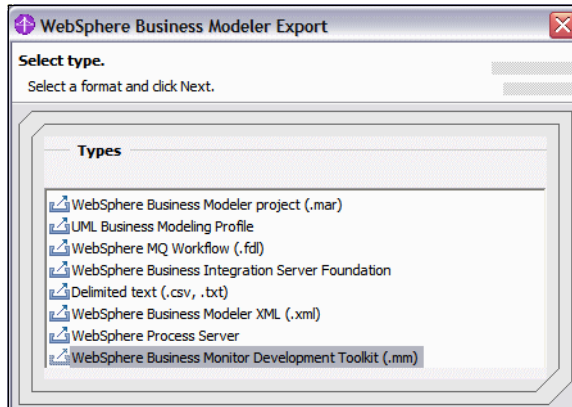


Figure 9-19 Export project with business measures

- ▶ Define the destination and source information export (Figure 9-20):
 - Target directory: C:\SG247148\sampcode\model\export
 - Project: *Clips and Tacks Future 1*
 - Select *Export entire project* or *Export specific elements* and select the *Order Handling (Future 1)* process.
 - Click *Finish*.

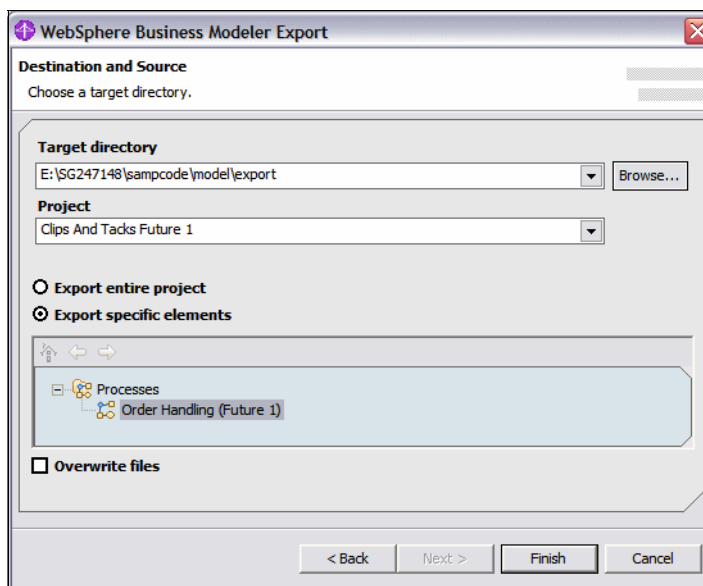


Figure 9-20 Export: Destination and source

Exported files

The target directory specified in Figure 9-20 contains three files:

- ▶ The two **.svg** files used for visualizing the process diagram:
 - `_Order_Handling__x0028_Future_1_x0029__Business_Measures_KM_Order_Handling__x0028_Future_1_x0029__KC.svg`
 - `_Order_Handling__x0028_Future_1_x0029__Business_Measures_MDM_Order_Handling__x0028_Future_1_x0029__MC.svg`
- ▶ One **.mm** file with the business measures:
 - `Order Handling (Future 1) Business Measures.mm`

We use the exported file in Chapter 12, “Developing and testing the business measures with the Monitor Toolkit” on page 327.

Exporting the process model for execution

If you have not exported the process model for execution as described in “Exporting the model for Integration Developer” on page 157, you can export the process now:

- ▶ Select either the project or the process and *Export* (context).
- ▶ Select the type of export you want. Select *WebSphere Process Server* and click *Next*.
- ▶ Specify the Target Directory and Process Server export details (refer to Figure 7-38 on page 159):
 - Target directory: `C:\SG247148\sampcode\model\export`
 - Project: `Clips and Tacks Future 1`
 - Select *Export entire project* or *Export specific elements* and select the *Order Handling (Future 1)* process.
 - Select *Module project name* and specify the name as `ClipsAndTacksF1`.
 - Select *Project interchange name* and specify the name as `ClipsAndTacksF1`.
 - Do not select *Enable default events*. This check box will generate events for all the activities. We rather delay event generation to the implementation in Integration Developer and select only the activities that are relevant for monitoring.
 - Click *Finish*.

Exported files

The target directory contains `C1 ipsAndTacksF1.zip`, which contains all the BPEL, WSDL interfaces, Java code, and so forth, used by WebSphere Integration Developer and WebSphere Process Server to interpret the business model created in Modeler. This file will be imported into Integration Developer to implement the application.

We use the exported project interchange file in Chapter 10, “Developing the application using WebSphere Integration Developer” on page 211.

Summary

In this chapter we described business measures in general and then added some business measures to the Order Handling (Future 1) process, so that we can monitor the business process using WebSphere Business Monitor.

We defined KPIs and measures to be calculated, and we added a dimension so that we can perform dimensional analysis of our numeric data (order counters and price) against the customer location.

We use this business measures model in Chapter 12, “Developing and testing the business measures with the Monitor Toolkit” on page 327, where we define in detail how the KPIs and measures are built using events, triggers, counters, and stopwatches.



Part 3

Development and testing

In Part 3 we describe how the process model exported from the Modeler is implemented using WebSphere Integration Developer.

In Modeler we indicated how certain tasks should be implemented; now we actually perform the implementation using business rules, Java code, a Web service call, and the human task manager. We also describe how to use the Business Rules Manager and how to implement security.

Finally we develop and test the business measures using the Monitor Toolkit and the Monitor Server test environment.



Developing the application using WebSphere Integration Developer

This chapter describes in detail how to use WebSphere Integration Developer to import the business process application exported from the Modeler and to complete the implementation.

WebSphere Business Modeler cannot be used to generate code for business rules, Java tasks, and to configure human tasks. These activities must be completed using a development tool.

We also have to implement the Web front-end that customers will use to submit their orders, and a database to store customer and order information.

We show how to configure and use the embedded WebSphere Process Server test environment to test the application.

Human tasks must be processed by the order manager and the shipper of ClipsAndTacks. We want to implement a nice user interface for these roles.

Note: The activities in this chapter are performed by the systems architect or integration developer, and possibly an application programmer.

Overview of the application implementation

To complete the application in WebSphere Integration Developer, we proceed through the following steps:

- ▶ Set up a database to keep the customer orders.
- ▶ Configure the WebSphere Process Server test environment.
- ▶ Import the model.
- ▶ Implement a business rule.
- ▶ Implement the Java activities.
- ▶ Configure the human tasks.
- ▶ Test the process.
- ▶ Add a stand-alone reference to invoke the process.
- ▶ Implement the Web front-end to invoke the process.
- ▶ Implement customized JSPs for the BPC Explorer.
- ▶ Run the application in the test environment.
- ▶ Implement an external Web service.
- ▶ Generate human task user interface.
- ▶ Implementing a human task application.
- ▶ Implement database access.
- ▶ Use the Business Rules Manager.

Setting up a database to keep the customer orders

The application requires a database to keep customer information, the products that can be ordered, and the orders of the customers.

The layout of the CLIPTACK database tables is shown in Figure 10-1.

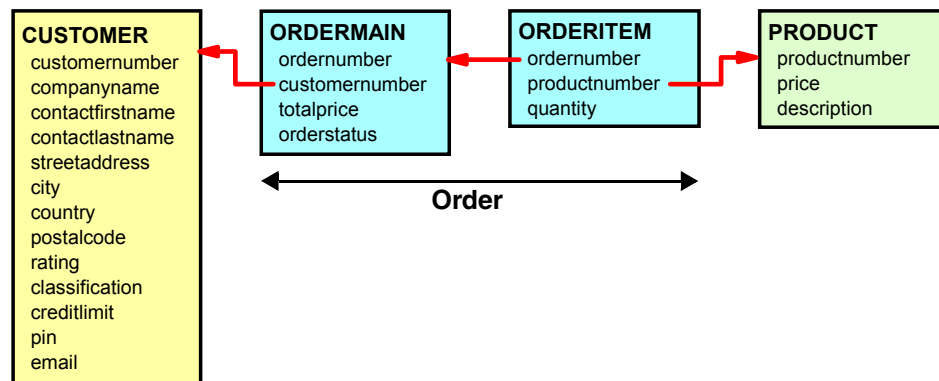


Figure 10-1 CLIPTACK database tables

Description of the database tables

The CLIPTACK database keeps track of the orders. Customers can submit orders through a Web front-end. During processing of the order, the status of the order is updated.

CUSTOMER table

A customer is a company that orders products from ClipsAndTacks. Most of the information about a customer is self-explanatory, so we will only document some of the columns:

- ▶ **Rating**—The rating column holds a numeric value that represents the credit rating of that customer. We assume that there is an external service that we can query to get a rating for a company.
- ▶ **Classification**—The classification of a customer is either REGULAR, SILVER, or GOLD. This column is not used in the first implementation of the application. Later, in Chapter 16, “Implementing the Future 2 process using WebSphere Integration Developer” on page 509, we implement business rules based on the classification.
- ▶ **Credit limit**—The credit limit column is used in the application to decide if a customer order is approved. We use the rating obtained from the external service to adjust the credit limit.
- ▶ **Pin**—the password used by a customer to login to the Web front-end application. For our sample the pin is identical to the customer number.

The CUSTOMER table is initialized with a few customers:

Number	Company	City, Country	Rating	CreditLimit
12345	ABC Finance Ltd.	Buffalo, USA	777	2000.00
11111	Auto Insurance Company	Etobicoke, Canada	666	1300.00
22222	ABC University	Markham, Canada	555	500.00
33333	US Auto Financing	Chicago, USA	765	1999.00
44444	Insurance For You	Toronto, Canada	632	1444.00

PRODUCT table

The PRODUCT table is simple and contains five products:

Number	Name	Price
RB-0001	All-In-One Printer	150.00
RB-0002	Manager Chair	79.00
RB-0003	5 MP Digital Camera	499.00
RB-0004	Cordless Phone with Answering Machine	89.00
RB-0005	3-Drawer File Cabinet	214.00

ORDERMAIN table

The ORDERMAIN table records an order of a customer. An order number is generated, and the order status is tracked:

- ▶ **Status**—The order status starts with the value NEW, then it is either APPROVED or DECLINED. Approved orders have their status 'changed to SHIPPED when processing is complete.
- ▶ **Total price**—The total price of an order is calculated from the items (products) that are ordered.

One sample order is initialized in the table:

Number	Customer	TotalPrice	Status
3001	12345	308.00	SHIPPED

ORDERITEM table

The ORDERITEM table contains the number of products that are ordered. The sample order contains two items:

Number	Product	Quantity
3001	RB-0001	1
3001	RB-0002	2

Populating the database

The Web front-end application (see “Implementing a Web front-end” on page 247) contains a servlet to defined the database tables and load the sample data. The data description language (DDL) statements and the sample data are retrieved from a properties file (`clipstacks.properties`).

Physical database

For our sample we use a Cloudscape™ database named CLIPTACK. We could have used DB2 as well. The access of the database is through a Cloudscape JDBC Provider (XA) data source. Once events are enabled on our application, we will require the two phase transaction support provided by XA data sources.

For the location of the database we use an existing directory inside the Process Server:

<WID-HOME>\pf\wps\databases	Test environment
<WPS-HOME>\profiles\ProcSrv01\databases	Production environment

WPS-HOME and WID-HOME are the installation directories of the WebSphere Process Server and Integration Developer.

Configuring the Process Server test environment

To access the CLIPTACK database from the server, we define a data source. We can use either the administrative console or a JACL script.

In this section we describe how to use the administrative console. When we deploy the application to a real server, we describe how to use a JACL script (see “Using a JACL script to define the data source” on page 404).

Starting Integration Developer

Start the Integration Developer using *Start* → *All Programs* → *IBM WebSphere* → *Integration Developer V6.0.2* → *WebSphere Integration Developer V6.0.2*.

We suggest to use a new workspace for all the work on the ClipsAndTacks application (Figure 10-2), for example:

```
C:\Documents and Settings\Administrator\IBM\wid6.0\workspace <== default  
C:\Workspaces\C1ipsAndTacks602 <== our choice
```

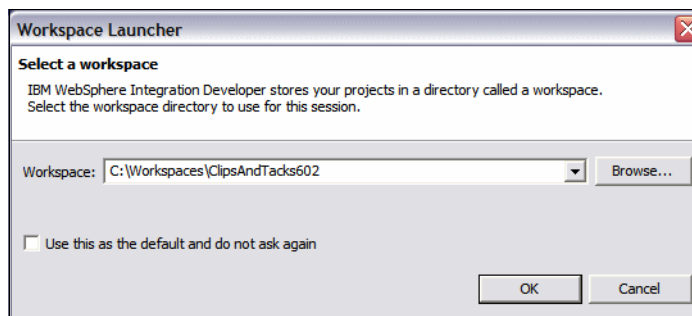



Figure 10-2 Workspace selection

Integration Developer opens with a Welcome page that you can close. The default perspective is the Business Integration perspective. For parts of our work we require the Web perspective.

Therefore, open the Web perspective (click the  icon, select *Other*, then select *Show all*, then select *Web*, and click *OK*), and when prompted, enable the Web development capability (Figure 10-3).

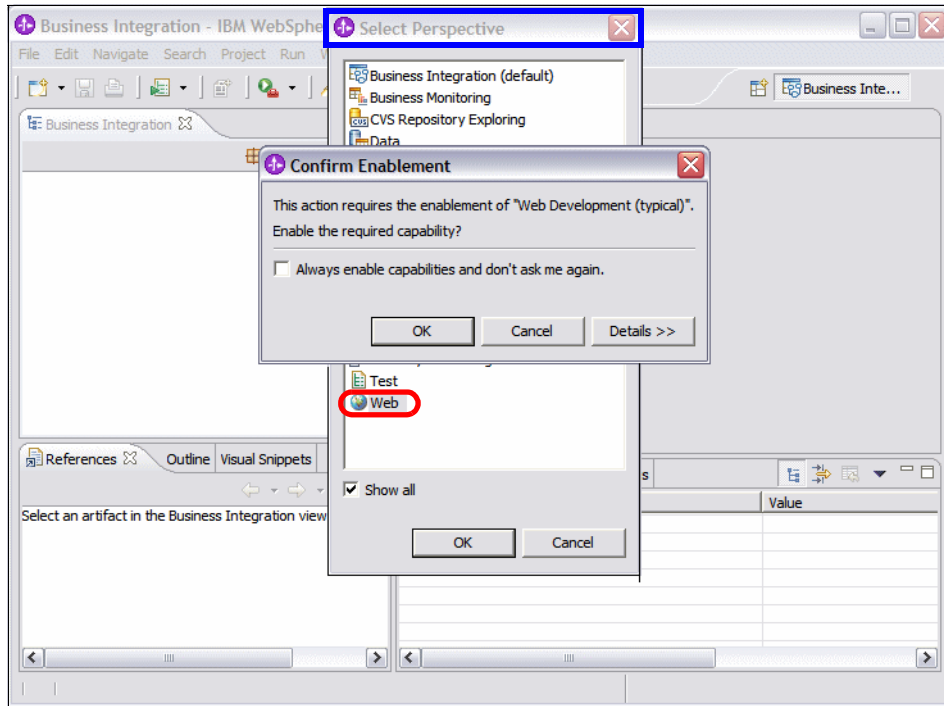



Figure 10-3 Opening the Web perspective

Starting the process server and administrative console

To define the data source we start the process server and administrative console:

- ▶ In the Web perspective, open the Servers view, select *WebSphere Process Server v6.0* and then click  to start the server (Figure 10-4). Wait until the server is started and the Console view shows the server console output. Be patient, starting the server takes about 5 minutes.

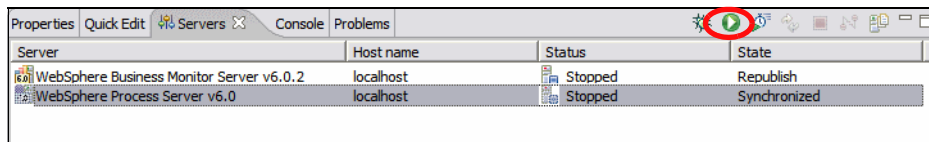


Figure 10-4 Start WebSphere Process Server v6.0

- ▶ Integration Developer communicates with the server through either RMI or SOAP. You can select the protocol by opening the server configuration (Figure 10-5)—double-click *WebSphere Process Server v6.0*.
- ▶ Select *Enable automatic publishing*.

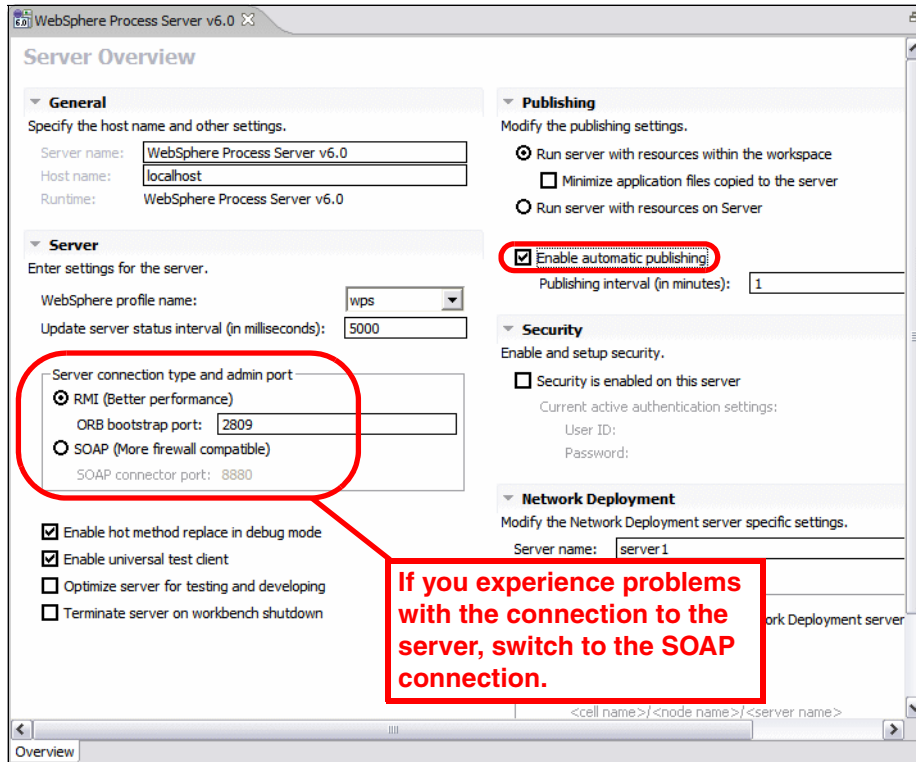


Figure 10-5 Process server configuration

- ▶ Start the administrative console by selecting the server and *Run administrative console* (context menu, Figure 10-6).

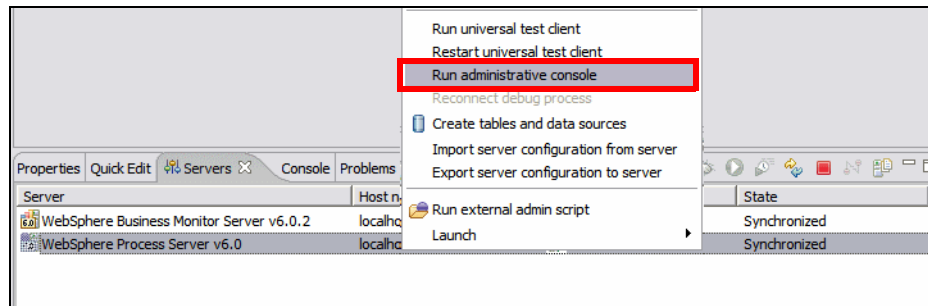


Figure 10-6 Start administrative console

- ▶ Enter any user ID or leave the field empty. A user ID is only required when security is enabled in the server. Click *Log in*.

Creating a data source for the database

Data sources are attached to a JDBC provider. We use the Cloudscape JDBC provider (XA) for our database:

- ▶ In the administrative console expand and select *Resources* → *JDBC Providers*.
- ▶ Select *Server* as scope, and click *Apply*. We define the data source for the CLIPTACK database at the server level (Figure 10-7).

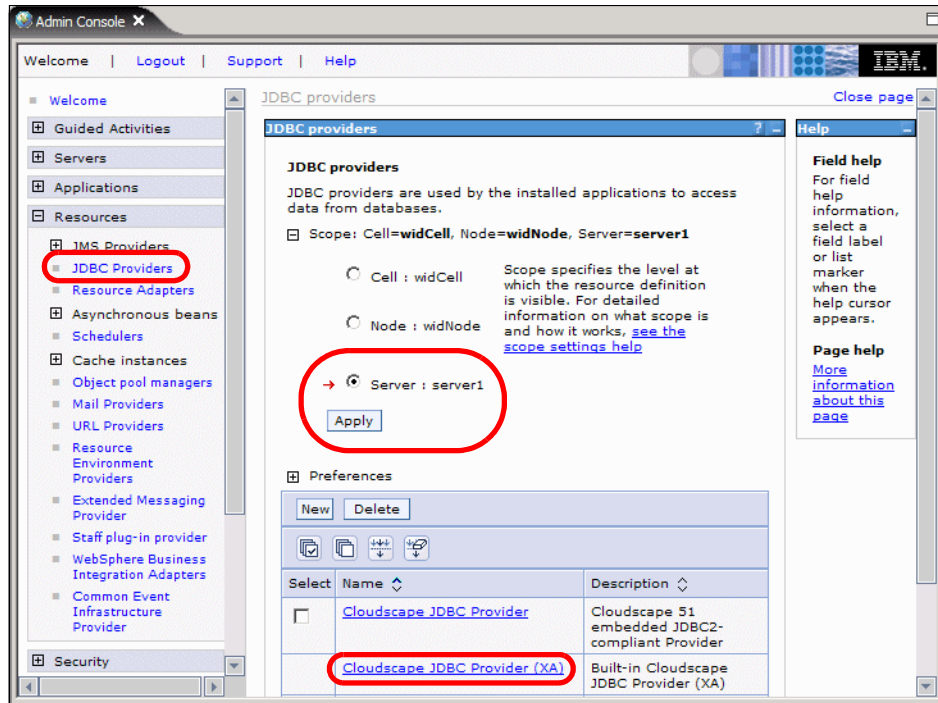


Figure 10-7 Change scope of JDBC Provider

- ▶ Click *Cloudscape JDBC Provider (XA)*.
- ▶ Click *Data sources* to define the data source (Figure 10-8).

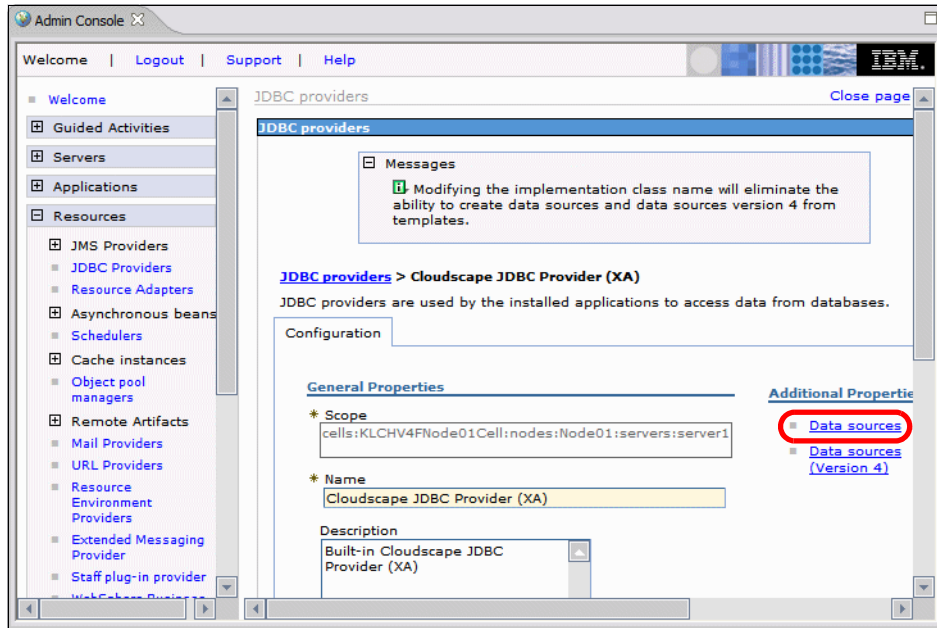


Figure 10-8 Cloudscape JDBC Provider (XA)

A number of Cloudscape data sources already exist (Figure 10-9). We create a new data source for the CLIPTACK database by clicking *New*.

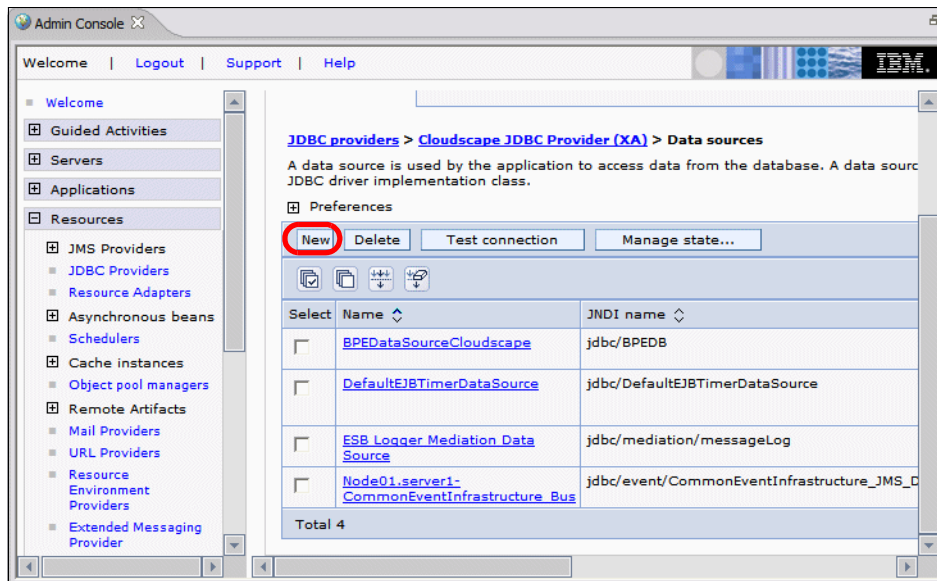


Figure 10-9 Cloudscape JDBC provider (XA) data sources

- ▶ Enter these values (Figure 10-10):
 - Name: ClipsAndTacks
 - JNDI name: jdbc/cliptack
 - Deselect: *Use this Data Source for container managed persistence*
 - Database name:
 - `${USER_INSTALL_ROOT}/databases/CLIPTACK`

The `${USER_INSTALL_ROOT}` variable points to the server profile where we want to define the database. In the test environment, this location is:

```
C:\<WID-HOME>\pf\wps
```
- ▶ Click *Apply*.

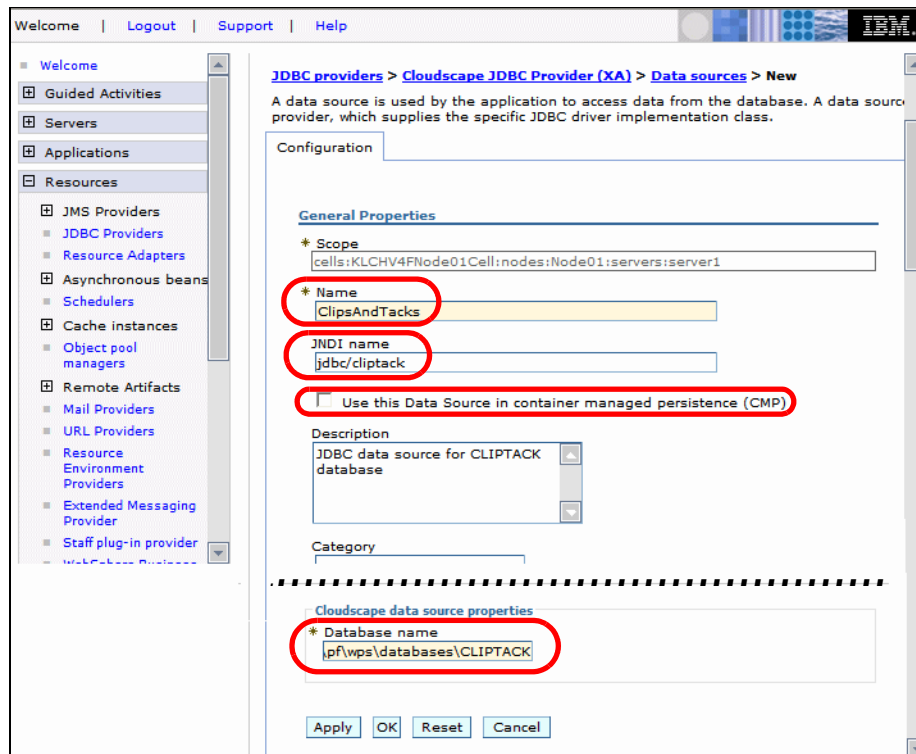


Figure 10-10 Create the data source for the CLIPTACK database

- ▶ Select *Custom properties* under Additional Properties.
- ▶ Select the *createDatabase* property and enter a value of *create*. Click *Apply*, then *OK*. This specification will create the database on the first access (Figure 10-11).

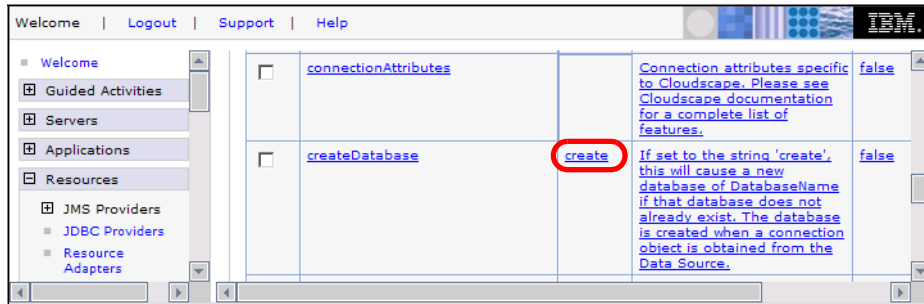


Figure 10-11 Automatic database creation

- ▶ Save the configuration (click *Save*).
- ▶ In the Cloudscape JDBC Provider (XA) data source list, select the *ClipsAndTacks* data source and click *Test connection* (Figure 10-12).

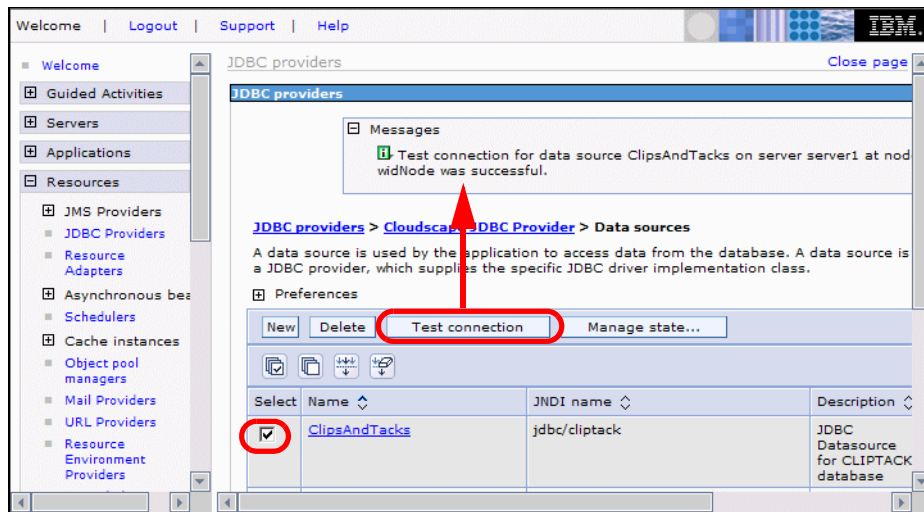


Figure 10-12 Testing the data source

- ▶ You should receive a message indicating success. You can verify that the database has been created by exploring the directory:


```
<WID-HOME>\pf\wps\databases\CLIPTACK
```
- ▶ Click *Logout* to close the administrative console.

At this time the database is created but does not contain any tables or sample data. We will initialize the database later using the Web front-end application (see “Initializing the database” on page 255).

Importing the model

In this section we import the model that was created in WebSphere Business Modeler and exported as a project interchange file.

We have two models that we can use:

- ▶ Select the model exported in Chapter 7, “Modeling the Future 1 business process” on page 121.
- ▶ Select the model exported in Chapter 9, “Defining KPIs and measures” on page 187 if you intend to use WebSphere Business Monitor to monitor the application. There is no difference between the files, unless you selected *Enable default events* when exporting the model—this enables events for all activities.

We provide the project interchange file in:

```
SG247148\sampcode\model\export\ClipsAndTacksF1.zip
```

If you intend to use the Monitor, you also have to import the .mm file exported in Chapter 9, “Defining KPIs and measures” on page 187, because the file contains the preliminary business measures definitions. However, this is discussed in Chapter 12, “Developing and testing the business measures with the Monitor Toolkit” on page 327.

Importing the project interchange file

To import the model and generate the processes in Integration Developer follow these steps:

- ▶ Select *File* → *Import*.
- ▶ Select *Project Interchange* and click *Next*.
- ▶ Click *Browse* and locate the ClipsAndTacksF1.zip file. Select the ClipsAndTacksF1 application and click *Finish* (Figure 10-13).

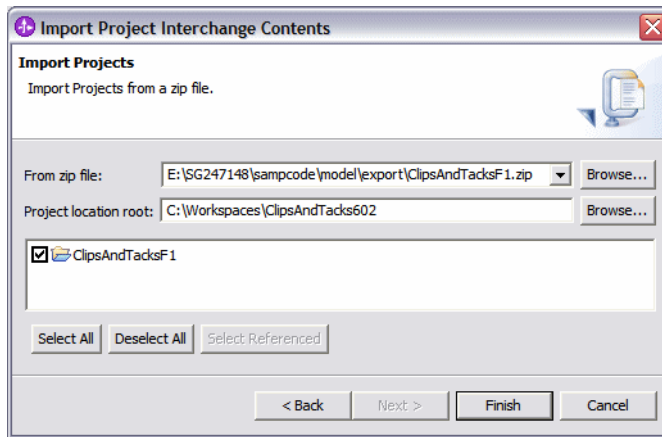
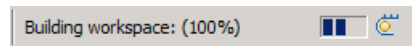


Figure 10-13 Import project interchange file

Be patient; from the model, a number of projects are generated and compiled. Watch the progress indicator at the bottom right:



Generated projects from the model

The list of projects that are generated from the model is shown in Figure 10-14.

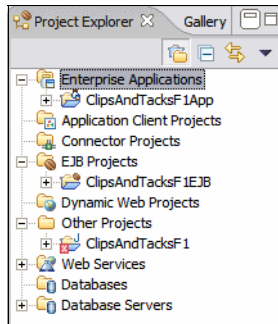


Figure 10-14 Projects generated from model import

- ▶ ClipsAndTacksF1—the business process, a Java project
- ▶ ClipsAndTacksF1App—the enterprise application
- ▶ ClipsAndTacksF1EJB—an EJB™ project with a number of session beans and one message-driven bean

Our work will be concentrated on the ClipsAndTacksF1 project. Note that it shows an error, which you can also see in the Problems view (Figure 10-15):

The definition of operation InputCriterion must have at least one enabled selection criterion -- either a criteria-based destination or a default destination.

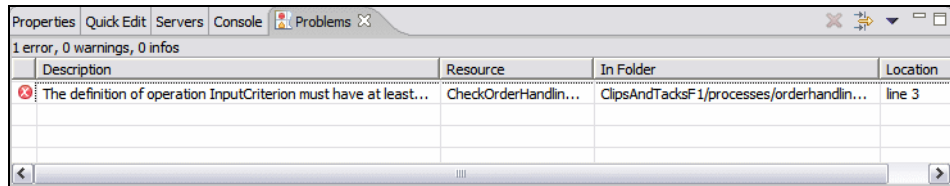


Figure 10-15 Error after import

When you expand the ClipsAndTacksF1 project in the Business Integration perspective, you can see that the error comes from the rule group CheckOrderHandlingPolicyforAutomaticApproval.

The business rule for automatic approval was an annotation in the Modeler; now we have to implement a real business rule (see “Implementing a business rule” on page 229).

Business Integration view

In the Business Integration view we can see the details of the business process (Figure 10-16).

Notice that all the activities have a suffix appended to the name and that blanks and parentheses have been eliminated.

The suffix is dependent on the name of the process. If you change the process name, different suffixes are generated.

The activity type that we selected in the Modeler has been used to generate different artifacts (components) for the development tool.

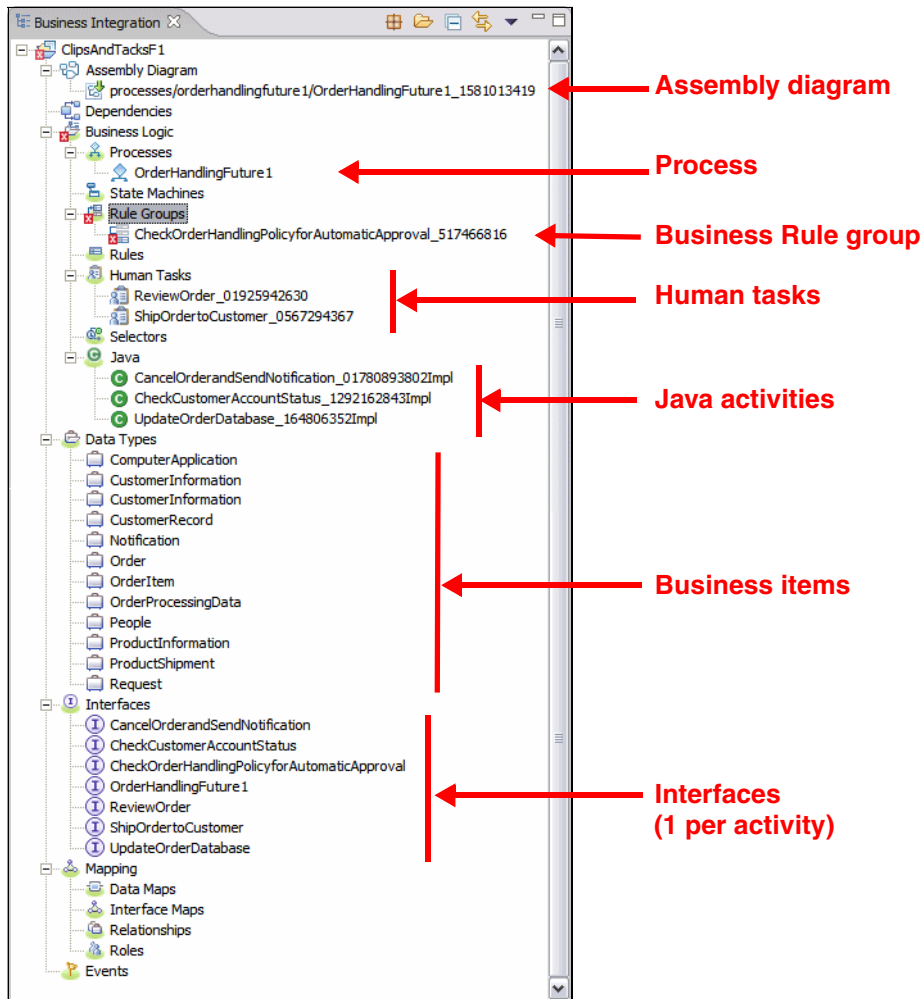


Figure 10-16 Business Integration view

Assembly diagram

The assembly diagram is shown in Figure 10-17. The diagram does not imply a sequence of operation of the six activities on the right; it only shows what activities are invoked by the process. Notice the different icons for the activities; we have one business rule, three Java implementations, and two human tasks. When you select an activity, some details are visible in the Properties view.

Right-click in the diagram and select *Show Display Name* to display the short names of the activities.

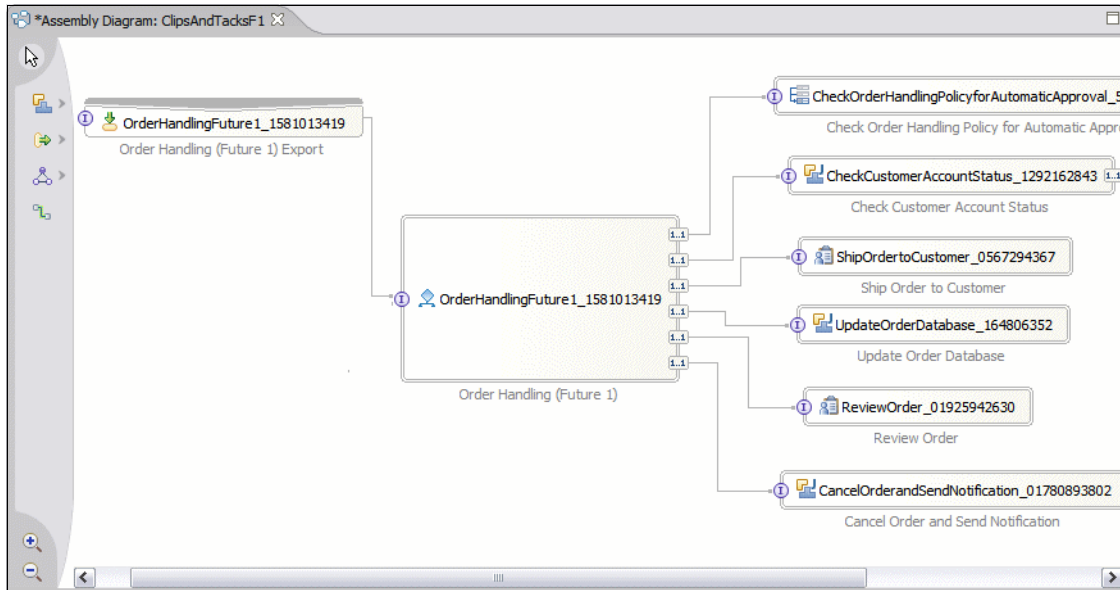


Figure 10-17 Assembly diagram

Process diagram

The process diagram for OrderHandlingFuture1 is shown in Figure 10-18.

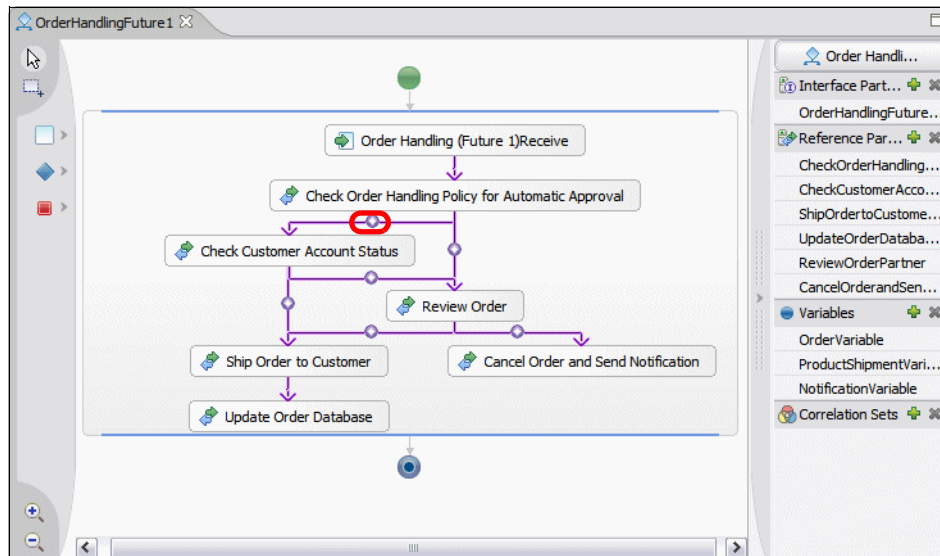


Figure 10-18 Process diagram

When you select one of the links (indicated by the square dots in Figure 10-18) you can see the corresponding code in the Properties view, Details tab:

```
return ((OrderVariable.getDataObject("ProcessingPreference")
        .getBoolean ("automaticApproval")) == (true));
```

Keeping processes once they are finished

The Properties view shows a number of details about the process. For example, in the Details tab for the process (Figure 10-19) you can see this information:

- ▶ The process is long running.
- ▶ Automatically delete the process after completion is selected by default.
You can deselect this flag so that finished processes are still visible, but only use this in the test environment and not for production. Save the process afterwards.
- ▶ The valid from date is the date of the export from Modeler. With version 6.0.1 of the BPM products it was important that we did not alter the valid from date in the Integration Developer if we were going to use the Monitor. This was because it was necessary to maintain consistency with the monitoring model file that was exported from Modeler for use by the Monitor. However, with the 6.0.2 versions of the products we can update the valid from date once the process is in the Integration Developer, because there is no longer an export directly from Modeler to Monitor, and we cannot set the date in Modeler.

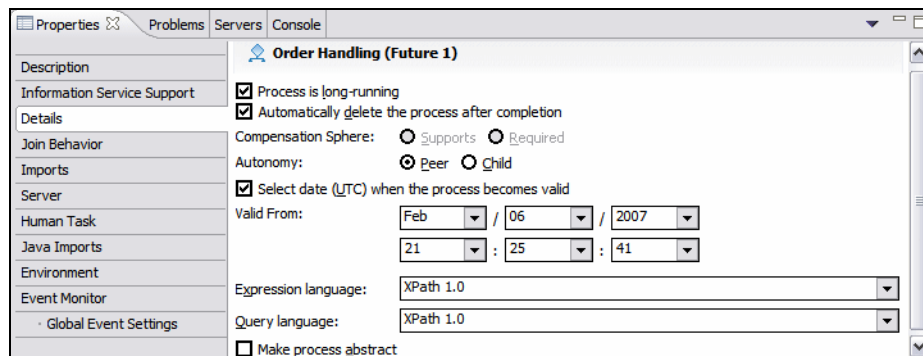


Figure 10-19 Process properties

Business objects (data types)

The data types are imported as business objects from the Modeler as a `BusinessItems.xsd` file under `ClipsAndTacksF1` → `businessitems`.

You can open a data type to see its definition in a diagram (Figure 10-20).

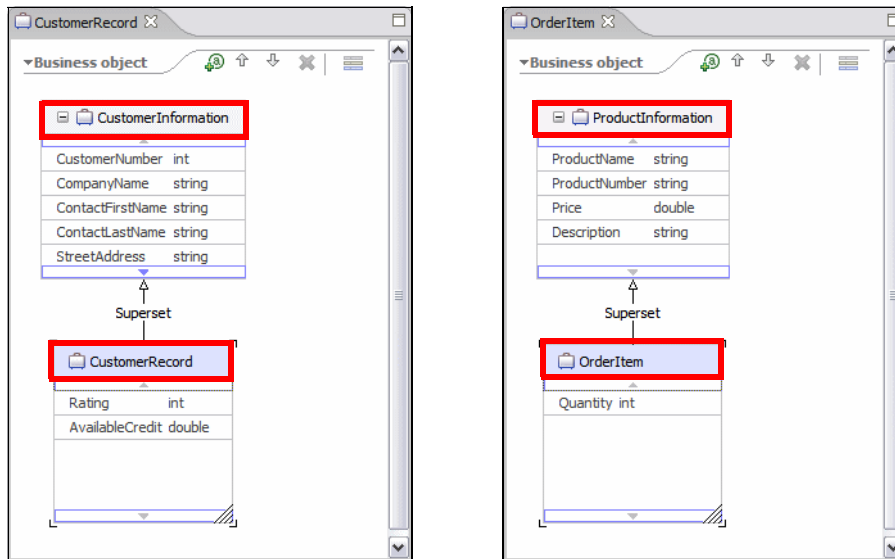


Figure 10-20 Data types

Physical resources

The Physical Resources view shows all of the file-level resources in the modules and libraries, so we can use this view to see all the files that make up the business process. Open the Physical Resources view (*Window* → *Show View* → *Physical Resources*) and review the contents:

- ▶ The business items folder with the BusinessItems.xsd file.
- ▶ The gen/src/processes/orderhandlingfuture1/brg folder with the generated Java code CheckOrderHandlingForAutomaticApproval.
- ▶ The processes/orderhandlingfuture1 folder with:
 - A business rule group
brg/CheckOrderHandlingPolicyforAutomaticApproval.
 - A wsdl file for each activity, which corresponds to each interface in the Business Integration view.
 - A component file with references to the interface and the implementation. There is a component file for each activity and one for the process.
 - A tel file and a _tel.mon file for each human task.
 - The OrderHandlingFuture1.bpel file. This is the business process itself; opening this file opens the process diagram.
 - A .java file for each Java activity
- ▶ The sca.module which is the assembly diagram for ClipsAndTacks.

You can also see the physical files in the Web perspective. Switch back to the Business Integration view.

The next tasks are to finish the implementation of the process.

Changing the namespace

Prior to implementing the business rule and other activities, we change the namespace for the data types. The Modeler always generates a namespace of `http://Businessitems`. Later, we will deploy another version of the process (ClipsAndTacksF2), where some data types are different. To be able to run both business process versions at the same time, we must ensure that the namespaces for the two processes are unique.

- ▶ In the Business Integration perspective, select the `CustomerInformation` business object and *Refactor* → *Change Namespace* (context menu).
- ▶ Click *OK* to agree to changing the namespace of the other business objects with the same namespace.
- ▶ Append `F1` to the current namespace so that the new namespace is `http://BusinessitemsF1` (Figure 10-21)

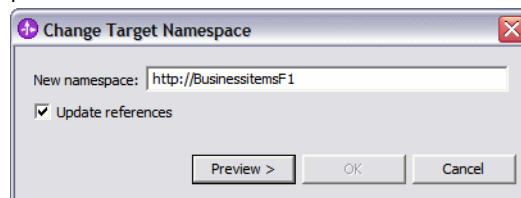


Figure 10-21 Change namespace

- ▶ Click *Preview* and then click *OK*.

Implementing a business rule

The business rule for automatic approval is an annotation in the Modeler and appears in the Integration Developer Properties view when the Check Order Handling Policy for Automatic Approval task is selected in the business process editor:

```
Order Handling Policy
=====
Default: Orders are reviewed by the system for automatic approval
=====
```

If the total price of an order is less than \$750,
then the order can be automatically approved without review.
=====

Implementing the business rule group

Next we need to implement the Check Order Handling Policy for Automatic Approval business rule group:

- ▶ Open the business rule group Check Order Handling Policy for Automatic Approval, either from the assembly diagram, or from the Business Integration perspective (expand *ClipsAndTacksF1* → *Business Logic* → *Rule Groups* → *CheckOrderHandlingPolicyforAutomaticApproval*).
- ▶ Select *InputCriterion* (Figure 10-22).

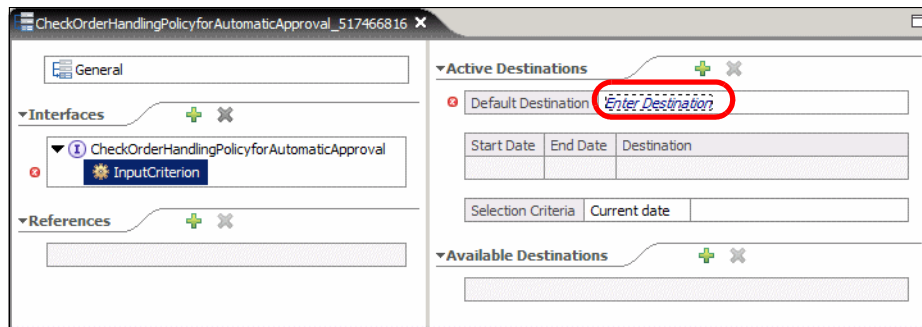



Figure 10-22 Rule group destination

- ▶ Click *Enter Destination* and select *New RuleSet* from the pull-down menu. In the New Rule Set enter these values:
 - Folder: *processes/orderhandlingfuture1* (default)
 - Name: *AutomaticApprovalF1*
- ▶ Click *Finish*.

The editor for the new rule set opens:

- ▶ Add an action rule by clicking the *Add Action Rule* icon .
- ▶ In the new rule, click *Action*, then select *Output*, =, *Input* (Figure 10-23). Press **Ctrl+Space** to use content assist.

Note: Another option would be **CopyBO(Input)** to make a copy of the input object.

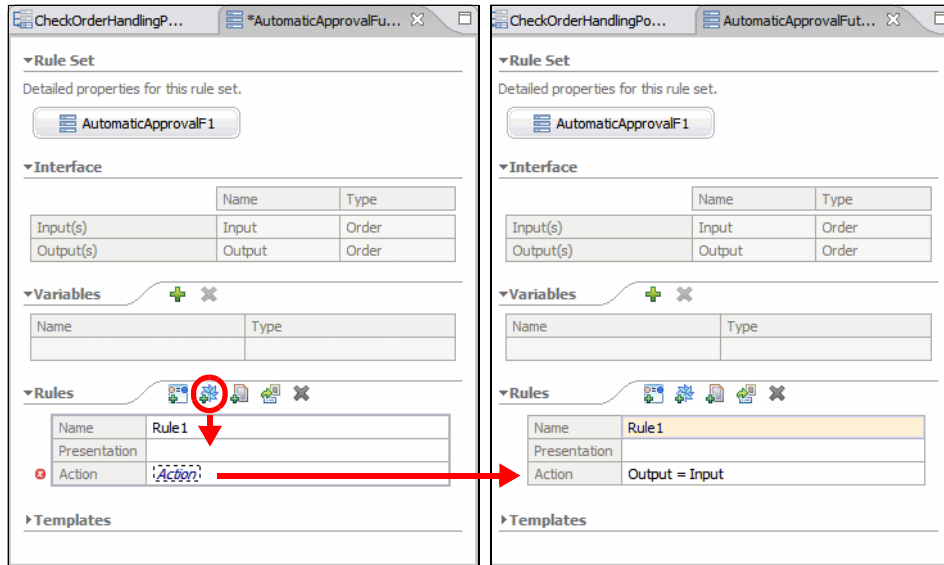



Figure 10-23 Defining business rules

- ▶ Save the new rule and the modified rule group. The error disappears from the Problems view.

Defining a business rule based on a template

The Business Rules Manager allows you to dynamically alter business rule parameters at runtime. To make use of the Business Rules Manager we must have at least one business rule that is based on a template. Later in the chapter (see “Using the Business Rules Manager” on page 295) we discuss how to use the Business Rules Manager.

- ▶ In the Templates section, click the *Add If then Template* icon . A template is added. Change the name to Total Price Template (Figure 10-24).

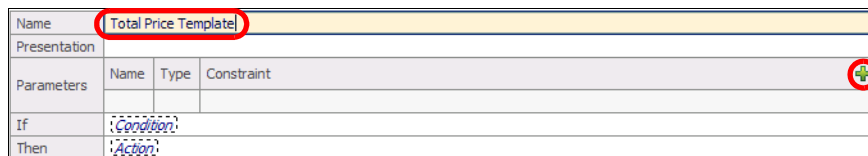



Figure 10-24 Business rule template

- ▶ Click the  icon to add a parameter (Figure 10-25):
 - Change the parameter name to totalPrice.
 - Click in the Type field and select *double*.

- Click *None* (under Constraint) and select *Range*. An expression is added.
- Click *Enter Expression* and select *Exclusive Range*.
- Click *number* in both cases and enter 700.00 and 800.00.

Name	Total Price Template		
Presentation			
Parameters	Name	Type	Constraint
	totalPrice	double	Range > 700.00 and < 800.00
If	:Condition:		
Then	:Action:		

Figure 10-25 Business rule template: Parameter

- ▶ Click *Condition* (If) and select *Input.Totalprice <= totalPrice*.
- ▶ Click *Action* (Then) and select *Output.ProcessingPreference.automaticApproval = true*.
- ▶ For the Presentation field enter:


The maximum order value for automatic approval is

Click after the text and an arrow shows up. Click the arrow and select *totalPrice*.

The final template rule is shown in Figure 10-26.

Name	Total Price Template		
Presentation	The maximum order value for automatic approval is :totalPrice:		
Parameters	Name	Type	Constraint
	totalPrice	double	Range > 700.00 and < 800.00
If	Input.TotalPrice <= totalPrice		
Then	Output.ProcessingPreference.automaticApproval = true		

Figure 10-26 Business rule template: Complete

- ▶ In the Rules section click the *Add Template Rule* icon . Select *Total Price Template*. A second rule is added (Figure 10-27).

Name	Rule2
Template	Total Price Template
Presentation	The maximum order value for automatic approval is <u>Enter Value</u>

Figure 10-27 New business rule using a template

- ▶ Overtyping *Enter Value* with 750.00 (Figure 10-28). Orders with a total price lower than \$750.00 are automatically approved. However, we have the facility to alter this limit at runtime, as long as the new limit is between \$700.00 and \$800.00. We will illustrate this in “Using the Business Rules Manager” on page 295.

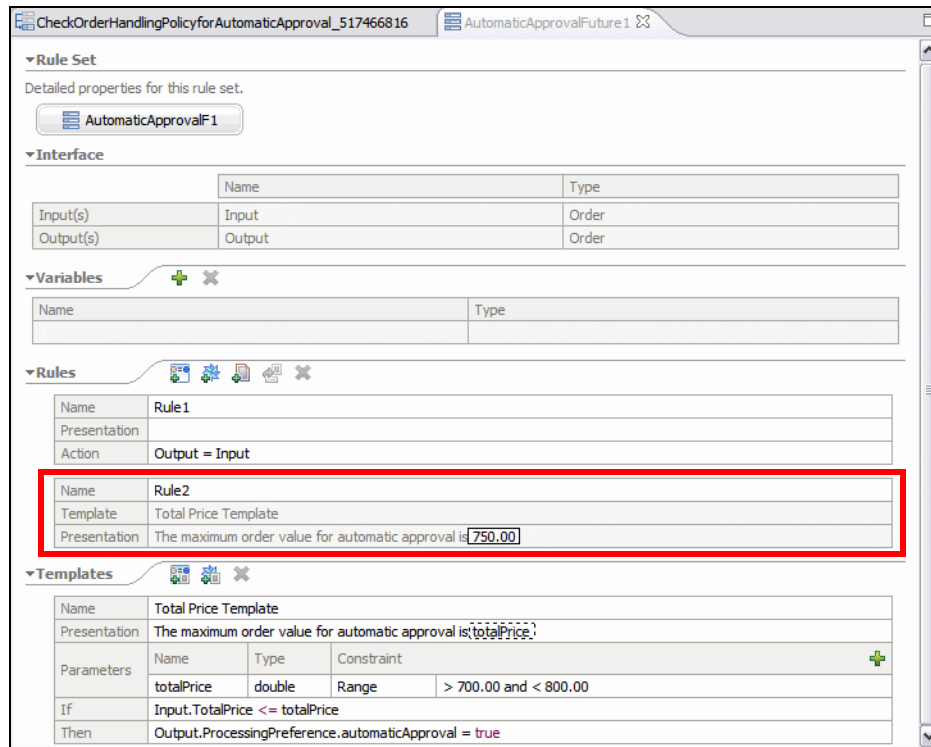


Figure 10-28 Final business rule using a template

Implementing the Java activities

We have three Java activities to implement, Check Customer Account Status, Update Order Database, and Cancel Order and Send Notification.

Check Customer Account Status

Open the Check Customer Account Status activity from the assembly diagram or from the Business Integration view (expand *ClipsAndTacksF1* → *Business Logic* → *Java*).

The Java code `CheckCustomerAccountStatus_1292162843Impl` opens. There is one method that requires code:

```
public DataObject InputCriterion(DataObject Input) {
    //TODO Needs to be implemented.
    return null;
}
```

The purpose of this activity is to verify the available credit of the customer so that a decision can be made for approval of the order. The available credit is based on the *rating* of the customer by some credit rating service.

For our scenario we use a simple approach:

- ▶ For the first implementation we use a JavaBean to randomly calculate a new rating, and then adjust the available credit.
- ▶ Later, in “Implementing an external Web service” on page 267 we use an external Web service to provide a new rating, and then adjust the available credit.

The logic for this activity is provided in a JavaBean with the name `CreditRating`. Complete the `InputCriterion` method with this code:

```
public DataObject InputCriterion(DataObject Input) {
    System.out.println("Check Customer Account Status Invoked");

    // create CreditRating bean
    com.clipstacks.credit.CreditRating creditRating =
        new com.clipstacks.credit.CreditRating();

    // call CreditRating bean to update the BO
    DataObject orderOut = creditRating.calculateCreditRating(Input);

    return orderOut;
}
```

Save the code. Ignore the error for now; we have to import the `CreditRating` JavaBean.

Implementing the credit check

To implement the credit check using the `CreditRating` JavaBean, follow these steps:

- ▶ Create a Java package named `com.clipstacks.credit` in the `ClipsAndTacksF1` project (select *File* → *New* → *Other* → *Java* → *Package*).
- ▶ Import the `CreditRating` class from the sample code into the package and the error should disappear:

```
SG247148\sampcode\wid\creditRating\CreditRating.java
```

Open the `CreditRating` code (use the Physical Resources view or the Web perspective) and study the code:

- ▶ A boolean switch, `useWebservice`, determines if an external Web service is invoked. For now, this value is `false`.

- ▶ A creditAdjustmentFactor (0.1) is used for the calculation of the new available credit of the customer.
- ▶ The calculateCreditRating method receives the input business object as parameter and returns the output business object.
- ▶ The customer information is retrieved from the input business object.
- ▶ For now, we will not investigate how the external Web service is called.
- ▶ A new rating is calculated as a random number between 500 and 800:


```
newRating = 650 + (new java.util.Random()).nextInt()%150;
```
- ▶ The available credit is calculated as:


```
availCredit = availCredit +(newRating-oldRating)*creditAdjustmentFactor;
```

This calculation adjusts the available credit up or down depending on the change in the rating.
- ▶ The output business object is created using the business object factory and copied from the input business object using the copy service.
- ▶ The customer rating and available credit are updated with the new values.
- ▶ The output business object is returned.

We will replace the random rating with an external Web service later.

Cancel Order and Send Notification

Open the Cancel Order and Send Notification activity. The Java code CancelOrderandSendNotification_01780893802Impl opens. Again, there is one method that requires code:

```
public DataObject InputCriterion(DataObject Input) {
    //TODO Needs to be implemented.
    return null;
}
```

Complete the code using the sample code in:

SG247148\sampcode\wid\codesnippets\cancelOrder.txt

- ▶ Add four variables to the class:

```
private com.ibm.websphere.sca.ServiceManager    serviceManager = null;
private com.ibm.websphere.bo.BOFactory         boFactory       = null;
private com.ibm.websphere.sca.Service         service          = null;
String namespace = "http://BusinessItemsF1";
```

- ▶ Add two lines to the constructor:

```
public CancelOrderandSendNotification_xxxxxxxImpl() {
    super();
```

```

        serviceManager = new com.ibm.websphere.sca.ServiceManager();
        boFactory = (com.ibm.websphere.bo.BOFactory)serviceManager
            .locateService("com/ibm/websphere/bo/BOFactory");
    }

```

- Complete the InputCriterion method (Example 10-1):

Example 10-1 Cancel Order and Send Notification Java implementation

```

public DataObject InputCriterion(DataObject Input) {
    System.out.println("Cancel order invoked");
    // retrieve customer e-mail address
    DataObject customer = Input.getDataObject("Customer");
    String emailAddress = customer.getString("Email");

    // create e-mail text
    String text1 = "Shipment for order: " + Input.getInt("OrderNumber") + "\n";
    String text2 = "Dear " + customer.getString("ContactFirstName") + " " +
        customer.getString("ContactLastName") + "\n";
    String text3 = "We are sorry that your order was cancelled.\n";
    String text4 = "The amount of $" + Input.getDouble("TotalPrice") +
        " was too much at this time\n";
    String text5 = "We hope to serve you again in the future.\n";
    List orderitems = Input.getList("OrderItems");
    int nrofitems = orderitems.size();
    String itemtext[] = new String[nrofitems];
    String text6 = "";
    for (int i=0; i<nrofitems; i++) {
        DataObject item = (DataObject)orderitems.get(i);
        String productName = (item.getString("ProductName") + "
            ").substring(0, 40);
        itemtext[i] = item.getInt("Quantity") + " " +
            item.getString("ProductNumber") + " "
            + productName
            + " $" + new BigDecimal(item.getDouble("Price"));
        text6 = text6 + itemtext[i] + "\n";
    }
    String emailText = "\n" + text1 + text2 + text3 + text4 + text5 + text6;

    // build notification data object
    DataObject notification = boFactory.create(namespace, "Notification");
    notification.setString("email", emailAddress);
    notification.setString("text", emailText);
    System.out.println("Cancel order email address: " + emailAddress);
    System.out.println("Cancel order email text: " + emailText);

    return notification;
}

```

Finally, add these two import statements to correct the problems by resolving the `List` and `BigDecimal` types:

```
import java.math.BigDecimal;
import java.util.List;
```

The purpose of the `Cancel Order` and `Send Notification` activity is to populate the notification data object with the customer's e-mail address and an e-mail text that informs the customer that their order has been cancelled. We print the contents of the e-mail to the console. Later in the chapter we will add database access to this activity so that the order database is updated to reflect the change in order status when an order is cancelled (see "Using JDBC in a Java activity" on page 292).

Update Order Database

Open the `Update Order Database` activity and review the `UpdateOrderDatabase_164806352Impl` Java class. Again, we have an `InputCriterion` method that has to be implemented:

```
public DataObject InputCriterion(DataObject Input) {
    //TODO Needs to be implemented.
    return null;
}
```

Complete the `InputCriterion` method with this code:

```
public DataObject InputCriterion(DataObject Input) {
    System.out.println("Update Order Database invoked");

    return Input;
}
```

Later in the chapter, this activity will be used to update the order database to reflect the fact that the order has been shipped (see "Using an information service" on page 287). At present, the method prints a message to the console indicating that the activity method has been invoked.

Configuring the human tasks

We have two human tasks that have to be configured for processing. As a first implementation, we define that everybody is allowed to process the human tasks, and we use the `BPC Explorer` that comes with the `Process Server` to work on the human tasks.

To configure the human tasks, perform these steps for both human tasks (Review Order and Ship Order to Customer):

- ▶ Open the human task from the assembly diagram or from the Business Integration view (expand *ClipsAndTacksF1* → *Business Logic* → *Human Tasks*).
- ▶ Select *Potential Owner* under Receiver Settings. In the Properties view, select *Everybody* as the staff group (Figure 10-29).

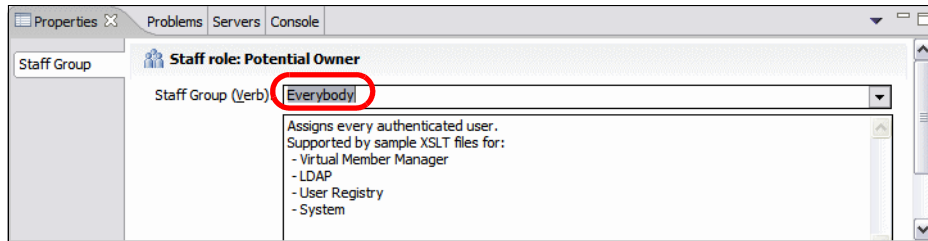



Figure 10-29 Human tasks: Change Role Members to Everybody

- ▶ For Client settings, click the *BPC Explorer* icon  to set up Business Process Choreographer Explorer for human tasks processing (Figure 10-30). Notice the Client Settings in the Properties view. We will defined tailored JSPs for human tasks later in “Implementing customized JSPs for the BPC Explorer” on page 257.

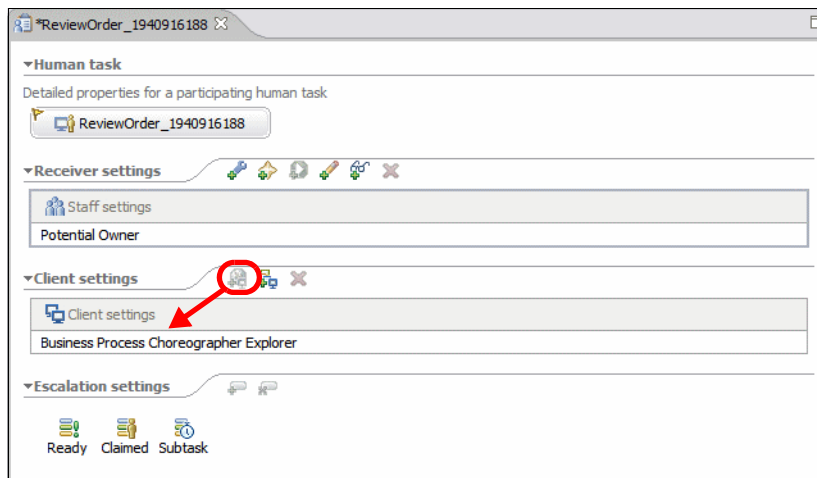


Figure 10-30 Human tasks configuration

Do not forget to configure both human tasks.

Testing the application

To test the application, we add it to the server and use the built-in process test facility.

Deployment of the application for testing

Before we can test the process, we have to deploy the application to the test server:

- ▶ In the Servers view, select the WebSphere Process Server v6.0 and *Add and remove projects* (context menu).
- ▶ Select the `ClipsAndTacksF1App` project and click *Add >* (Figure 10-31). Click *Finish* and the application is deployed. Watch the progress indicator in the bottom right of the Integration Developer window.

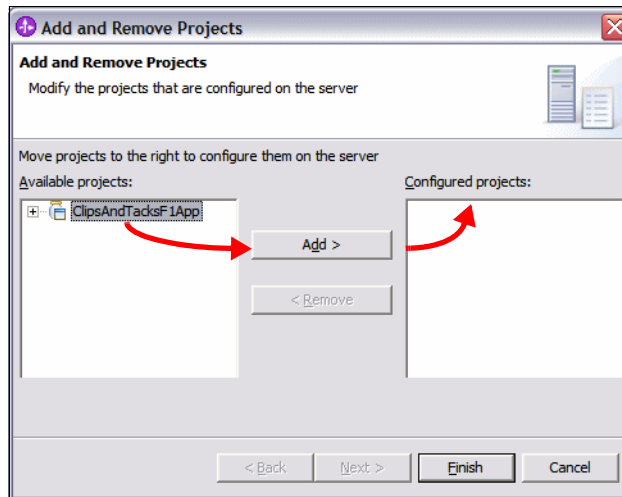


Figure 10-31 Add project to server for testing

Using the built-in process test facility

To test the `ClipsAndTacksF1` module, follow these steps:

- ▶ From the Business Integration perspective, select the `ClipsAndTacksF1` project and *Test* → *Test Module* (context menu). The `ClipsAndTacksF1_Test` panel opens (Figure 10-32).

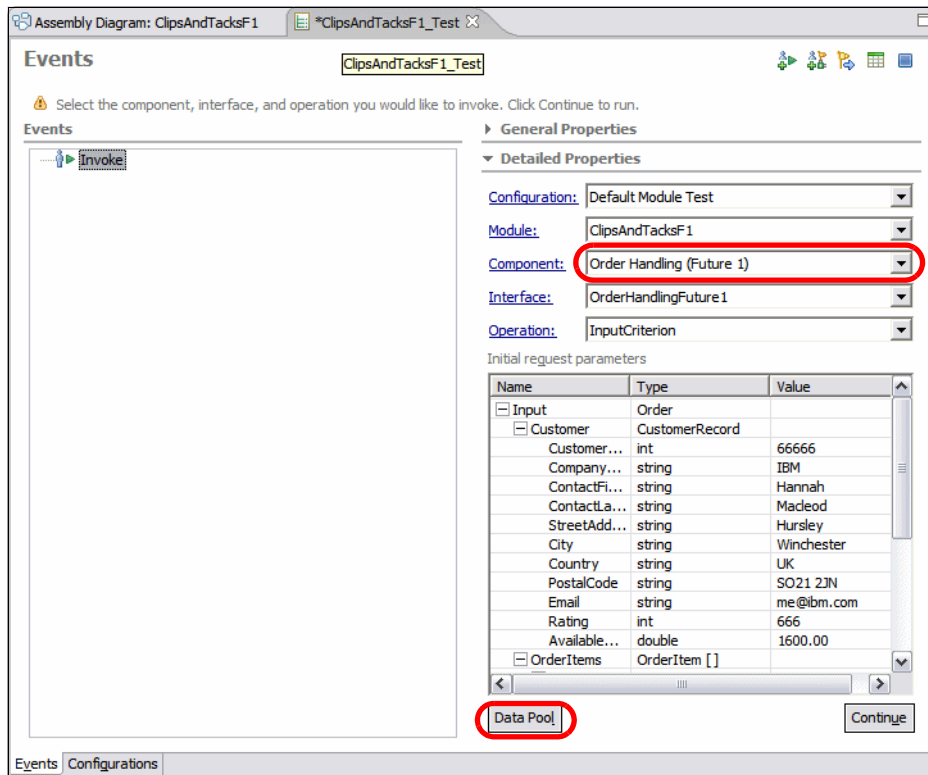


Figure 10-32 Testing: Input data

- ▶ Under Detailed Properties, change the Component to Order Handling (Future 1). By default it is set to the first activity in alphabetical order.
- ▶ Enter values for the Customer data.
- ▶ To enter order items, select OrderItems and *Add Element* (context menu).

This is quite some work. To reduce the effort for future tests, we can save the values in a data pool:

- ▶ Select *Input* (the first line) and *Add Value to Pool* (context menu). In the Value Name dialog, enter a suitable name, for example, CustomerIBM and click *OK*.
- ▶ Click *Continue*.
- ▶ Select the deployment location as *WebSphere Process Server v6.0* and click *Finish* (Figure 10-33).

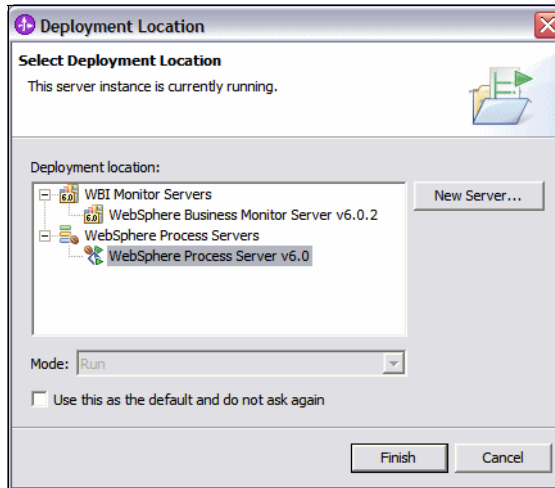


Figure 10-33 Testing: Deployment location

The process starts and events are displayed in the events list (Figure 10-34).

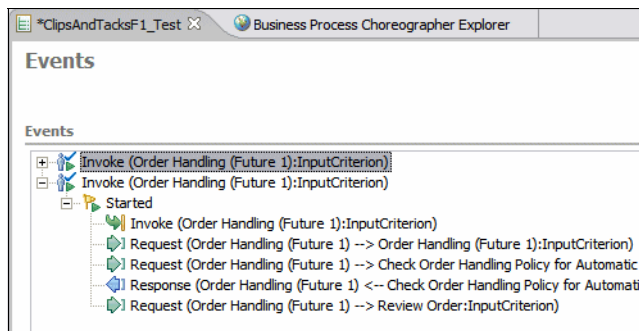


Figure 10-34 Testing: Process events

We can see that the Order Handling (Future1) process was invoked and has proceeded through the Check Order Handling for Automatic Approval activity (business rule) and is now waiting in the Review Order activity (human task).

Processing the human task using the BPC Explorer

Now we have to play the part of the order manager and decide if this order is approved or declined. For this purpose we start the BPC Explorer:

- ▶ Select the server in the Servers view and *Launch* → *Business Process Choreographer Explorer* or use an external browser with the URL:

`http://localhost:9080/bpc`

- ▶ The BPC Explorer opens with the My Tasks view and the Review Order activity is visible (Figure 10-35).

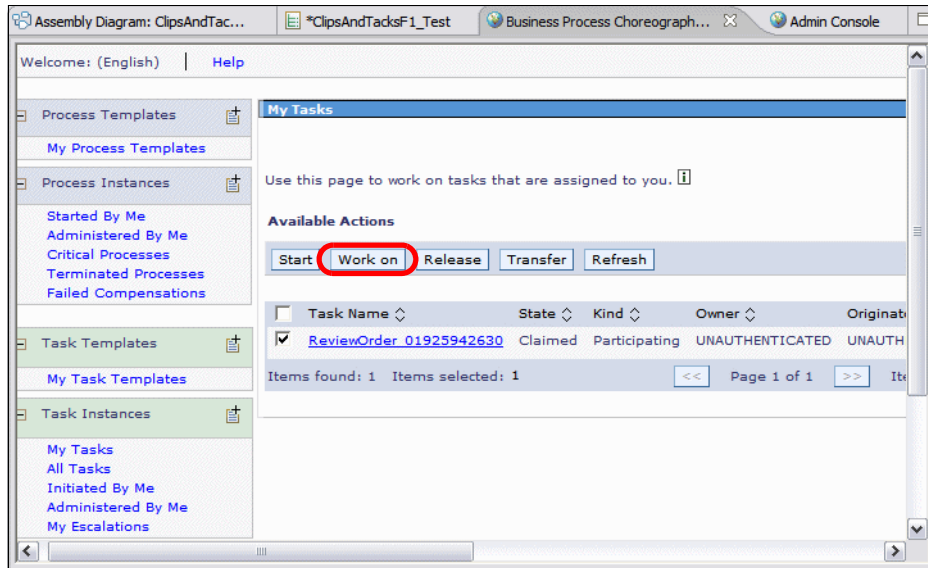


Figure 10-35 BPC Explorer: Tasks list

- ▶ You can click the task itself and look at the details and input and output messages. Note that the output message is empty. The human task must be worked on and the output message completed.
- ▶ To work on the task, select the task (check box) and click *Work on*. This action locks the task so that no other user can work on the task.
- ▶ The task opens with the input message filled in, and an empty output message (Figure 10-36):
 - The output message must be filled with data for further activities.
 - There is no simple way to do this, each field must be copied from the input or entered.
 - The most important field is the order status; it must be set to either APPROVED or DECLINED.
 - When the output message is ready, click *Complete*.
 - To delay the decision, you can click *Release* and the activity is available for processing once more.

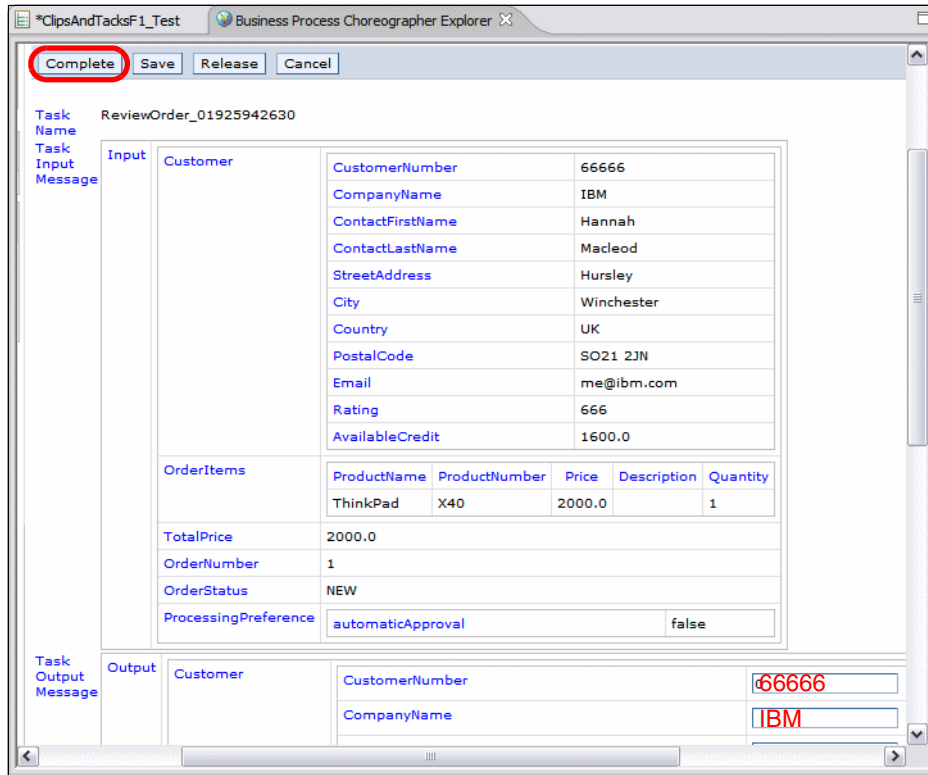


Figure 10-36 BPC Explorer: Work on

- ▶ Suppose that we approve the order and click *Complete*.
- ▶ Refresh the tasks list in the BPC Explorer (click *Refresh*) and you can see that process is now waiting in the Ship Order to Customer activity.
- ▶ Select the activity and click *Work on*.
- ▶ Enter the values for the output message (a packing slip number), then click *Complete*.
- ▶ The process finishes, which can be seen in the test events list (Figure 10-37).

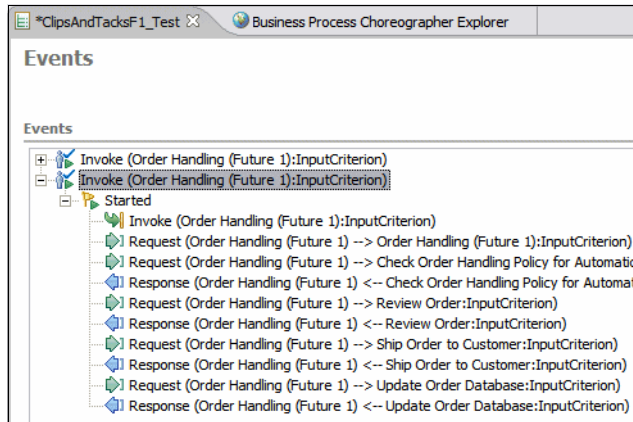


Figure 10-37 Testing: Process complete

We can see the process in the BPC Explorer by clicking *Started By Me* under Process Instances (Figure 10-38).

Note that finished processes are only visible if you deselect *Automatically delete the process after completion* in Figure 10-19 on page 227.

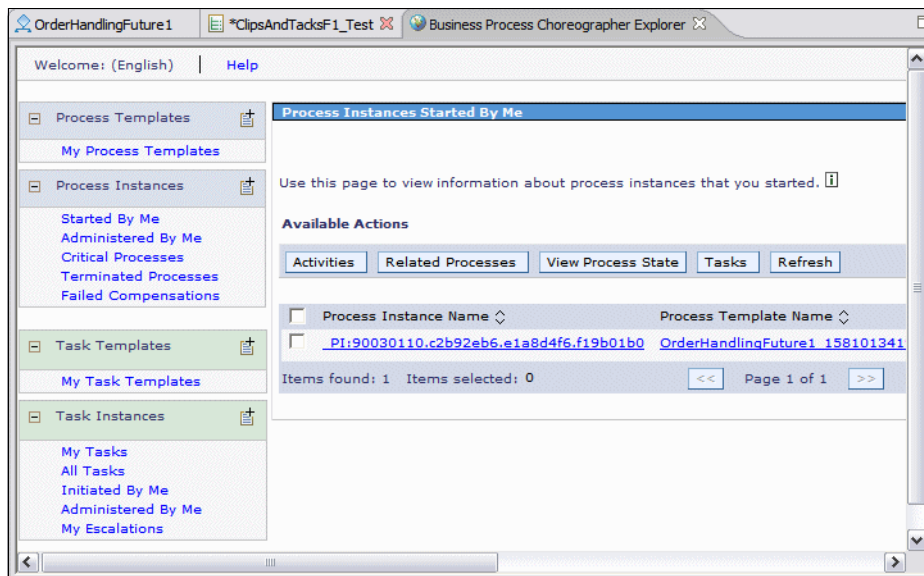


Figure 10-38 BPC Explorer: Complete processes

Click the process to see its input and output messages, as well as the list of activities, with an indication if they are finished or skipped (Figure 10-39).

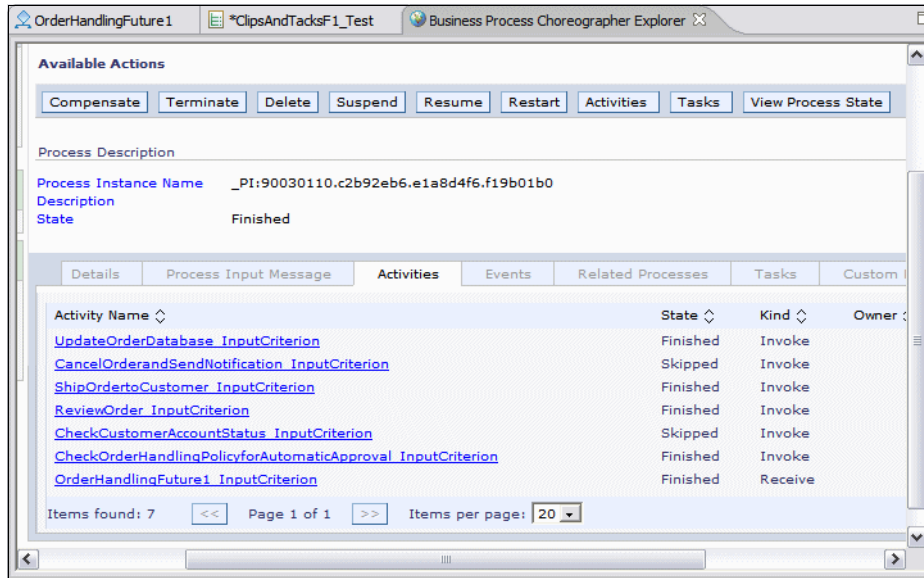


Figure 10-39 BPC Explorer: Completed activities

Rerunning tests and reusing data

To rerun a test, select the top event and *Rerun* (context menu). The same input data is used. For example, you can decline the order at this time.



For another test with different data, select the project and *Test* → *Test Module*. Do not forget to select the Order Handling (Future 1) component. To reuse the data from the data pool, select *Input* and select *Data Pool* (context menu). Select an existing entry, then modify the input data before clicking *Continue*.

For example, enter data with a total price below \$750.00 and the order is automatically approved. In the BPC Explorer the Ship Order to Customer activity is waiting and can be completed.

When closing the test, you are prompted to optionally save the execution trace. You can use any project location to save the execution trace.

Adding a stand-alone reference to invoke the process

Applications that are not defined as Service Component Architecture (SCA) components can interact with SCA components by using stand-alone references. Therefore, we define a stand-alone reference in the assembly diagram so that we can invoke the process from a Web front-end (JSPs):

- ▶ Open the assembly diagram.
- ▶ Click the arrow next to the *Import* icon , then select the *Stand-Alone References* icon , then click into the diagram (Figure 10-40).

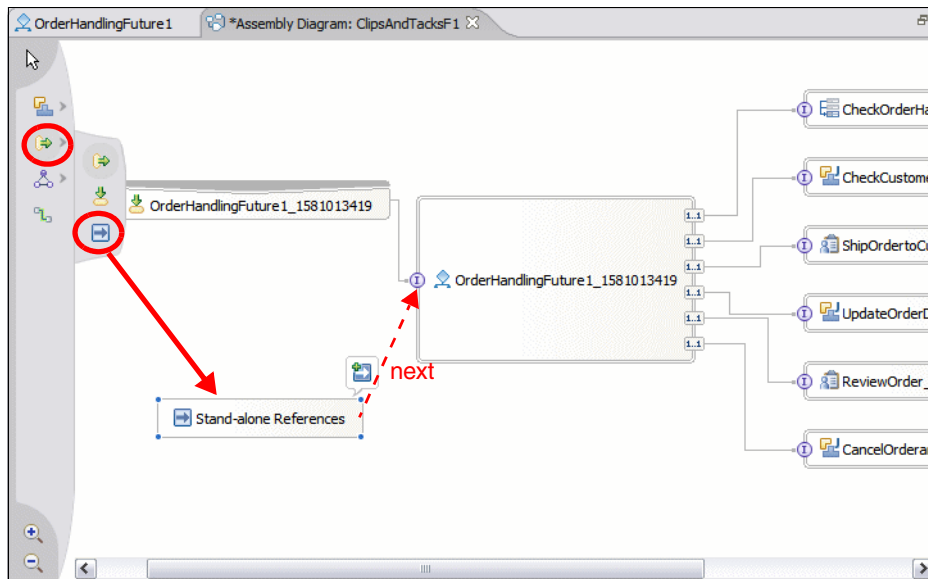

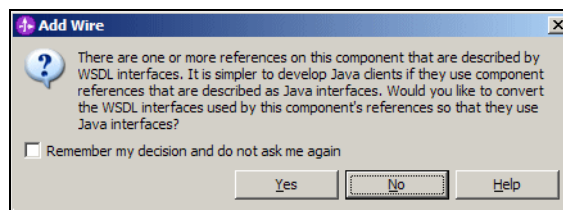


Figure 10-40 Adding a stand-alone reference

- ▶ Link the stand-alone reference to the Order Handling (Future 1) process. Select the wire icon , then click the reference and on the process:
 - A prompt appears to create a matching reference, click *OK*.
 - A prompt appears to create a Java interface instead of the WSDL interface. **Click No.** We will use the WSDL interface.



The stand-alone reference is named `OrderHandlingFuture1Partner` (look in the Properties view). We will use this name in the Web front-end,

- ▶ Rearrange the diagram (Figure 10-41).

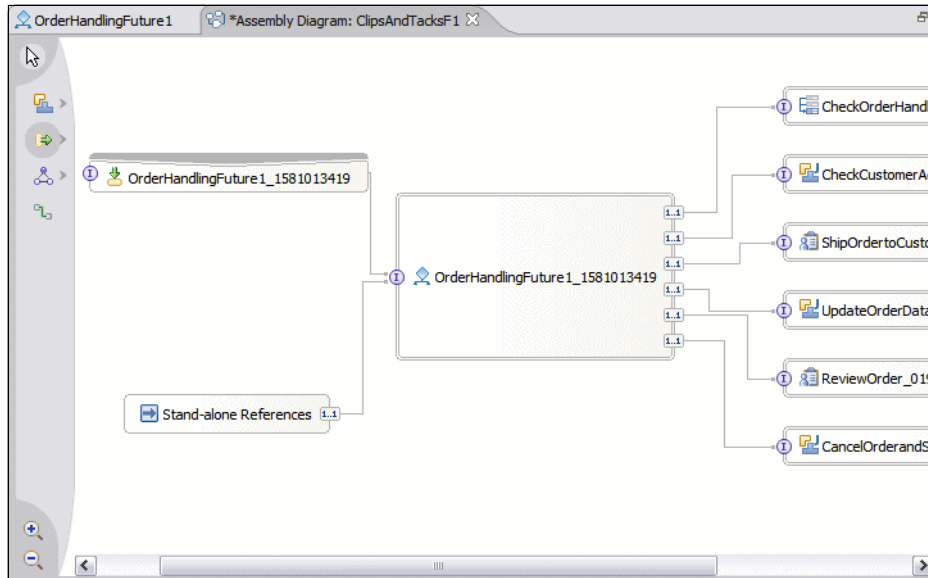


Figure 10-41 Process diagram with stand-alone reference

- ▶ Save and close the diagram.

Implementing a Web front-end

ClipsAndTacks uses a Web front-end application to submit customer orders to the business process. Through the self-service front-end, customers can select the products to be ordered and enter the order into the CLIPTACK database.

We do not describe here how to build such a Web application. We only describe the architecture and functionality, and how to add the Web application to the business process.

Architecture

The architecture of the Web application is shown in Figure 10-42:

- ▶ The home page is used for login of the customer. New customers can register themselves in the database.
- ▶ The customer then selects the products from a (small) catalog and submits the order.
- ▶ The order is stored in the database and the business process is invoked.

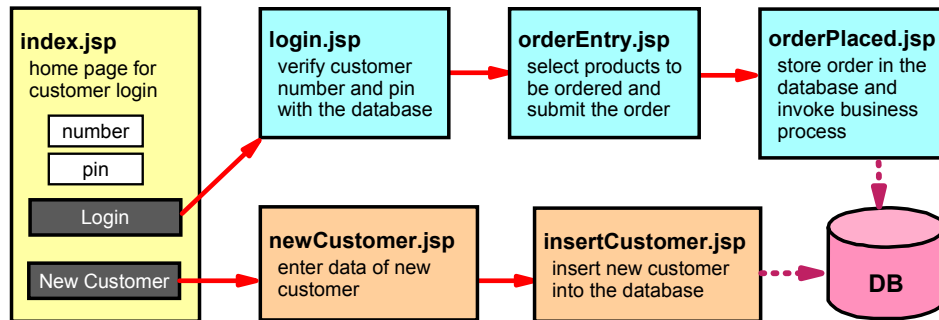


Figure 10-42 Web front-end architecture

Database access using the Java module

To access the database the Web application uses a data access object (DAO) and data transfer objects (DTO). The DAO contains the logic to access the CLIPACK database using JDBC. The DTOs hold the customer, product, and order information that is transferred between the Web application and the DAO.

- ▶ **CustomerDTO**—Holds all the customer information, matching the CUSTOMER table in the database.
- ▶ **ProductDTO**—Holds all the product information, matching the PRODUCT table.
- ▶ **OrderItemDTO**—Hold the information for one item ordered (quantity, product number and name, price).
- ▶ **OrderDTO**—Holds the complete order information with order number, total price, order status, automatic approval (set during the business process), one CustomerDTO, and an array of OrderItemDTO.
- ▶ **ClipsTacksDAO**—Used for all database access using the data source defined in the server. The DAO provides these methods:
 - **getCustomer**—Retrieve a customer and verify the pin.
 - **insertCustomer**—Insert a new customer into the database.
 - **deleteCustomer**—Delete a customer and all its orders from the database.
 - **updateCustomer**—Update the rating and credit limit of a customer (used by the business process).
 - **getProducts**—Retrieve all the products for order entry.
 - **insertOrder**—Insert a new order and its order items into the database.
 - **updateOrder**—Update the order status (used by the business process).
 - **getConnection**—Get a database connection using a data source reference. This method is called by all the access methods.

- **commit**—Commit changes to the database. This method is called by the methods that update the database.

Servlets

The Web front-end provides three servlets to initialize and manage the database:

- ▶ **CreateDatabaseServlet**—Runs the data description language (DDL) statements to define the tables and foreign key relationships, and to load the sample data (five customers, five products, one order). This servlet must be run before using the Web front-end. The servlet uses a properties file (`WEB-INF/clipstacks.properties`) that holds the SQL statements.
- ▶ **ListServlet**—Lists the content of the database.
- ▶ **DeleteServlet**—Deletes one customer from the database.

Properties file

The `clipstacks.properties` file contains the SQL statements to define the database tables and load sample data, enabling us to modify the data without reinstalling the application. Figure 10-43 shows an extract of the file.

```
database=CLIPTACK
createCustomer=CREATE TABLE CT.CUSTOMER ( customernumber INTEGER NOT NULL,
    companyname VARCHAR(32) NOT NULL, ..., PRIMARY KEY (customernumber) )
createProduct=CREATE TABLE CT.PRODUCT ( productnumber CHAR(10) NOT NULL,
    productname VARCHAR(40) NOT NULL, ..., PRIMARY KEY (productnumber) )
createOrder=CREATE TABLE CT.ORDERMAIN ( ordernumber INTEGER NOT NULL,
    customernumber INTEGER NOT NULL, totalprice DECIMAL(8,2) NOT NULL,
    orderstatus CHAR(8) NOT NULL, PRIMARY KEY (ordernumber) )
createItem=CREATE TABLE CT.ORDERITEM ( ordernumber INTEGER NOT NULL,
    productnumber CHAR(10) NOT NULL, quantity INTEGER NOT NULL, PRIMARY KEY
    (ordernumber, productnumber) )
orderCustomer=ALTER TABLE CT.ORDERMAIN ADD CONSTRAINT OrderCustomer FOREIGN
    KEY (customernumber) REFERENCES CT.CUSTOMER ON DELETE CASCADE
itemOrder=ALTER TABLE CT.ORDERITEM ADD CONSTRAINT OrderItemOrder FOREIGN KEY
    (ordernumber) REFERENCES CT.ORDERMAIN ON DELETE CASCADE
itemProduct=ALTER TABLE CT.ORDERITEM ADD CONSTRAINT OrderItemProduct FOREIGN
    KEY (productnumber) REFERENCES CT.PRODUCT ON DELETE CASCADE
insertCustomer=INSERT INTO CT.CUSTOMER VALUES (12345, 'ABC Finance Ltd.',
    'Marc', 'Shankaran', '1 Main Street', 'Buffalo', 'USA', '82840', 777,
    'GOLD', 2000.00, 12345, 'marc@abcfinance.com'), (11111, ...), ...
insertProduct=INSERT INTO CT.PRODUCT VALUES ('RB-0001', ...), ...
insertOrder=INSERT INTO CT.ORDERMAIN VALUES (3001, 12345, 308.00, 'SHIPPED')
insertItem=INSERT INTO CT.ORDERITEM VALUES (3001, 'RB-0001', 1), ...)
```

Figure 10-43 Properties file with SQL statements

Process invocation

The Web front-end contains a class, **InvokeOrderHandling**, that invokes the business process when an order is placed in the `orderPlaced.jsp`.

The business process is found using the name of the stand-alone reference (`OrderHandlingFuture1Partner`), then the operation (`InputCriterion`) is invoked using an input business object as parameter:

```
standAloneReferenceName = "OrderHandlingFuture1Partner";
operationName           = "InputCriterion";
namespace               = "http://BusinessItemsF1";

serviceManager = new com.ibm.websphere.sca.ServiceManager();
service = (Service)serviceManager.locateService(standAloneReferenceName);
operationType = service.getReference().getOperationType(operationName);

DataObject input = // create the business object from the orderDTO
Ticket output = service.invokeAsync(operationName, input);
```

To create the input business object, a factory (`BOFactory`) is used:

```
boFactory = (BOFactory)serviceManager
            .locateService("com/ibm/websphere/bo/BOFactory");
DataObject order = boFactory.create(namespace, "Order");
order.setInt("OrderNumber", orderDTO.getOrderNumber());
...
DataObject customer = boFactory.create(namespace, "CustomerRecord");
customer.setString("CompanyName", customerDTO.getCompanyName());
...
order.setDataObject("Customer", customer);
...
DataObject input = boFactory.createByType(operationType.getInputType());
input.setDataObject("Input", order);
```

The full code on how to invoke the business process is in the `execute` method of the `InvokeOrderHandling` class. The code also includes the code for using the Java interface instead of the WSDL interface (in comments).

Note: This logic uses the Service Component Architecture (SCA) API as provided by the IBM products. The business objects used in the business process are Service Data Objects (SDO). The specification of SCA and SDO are available at:

<http://www.ibm.com/developerworks/webservices/library/specification/ws-sca/>
<http://www.ibm.com/developerworks/library/specification/ws-sdo/>

For more information on SDO, consult the Redbooks publication, *WebSphere Studio 5.1.2 JavaServer Faces and Service Data Objects*, SG24-6361.

Importing the Java module

We provide a data access object (ClipsTacksDAO) and a number of data transfer objects (CustomerDTO, OrderItemDTO, OrderDTO, ProductDTO) that use JDBC to retrieve and update the CLIPTACK database.

This code is packaged into a JAR file that can be added to any enterprise application that has to access the database. A data source reference is required in the Web or EJB project that uses the database access code.

To import the Java project containing the database access code, follow these steps:

- ▶ Select *File* → *Import*.
- ▶ Select *Project Interchange* and click *Next*.
- ▶ Click *Browse* and locate the ClipsAndTacksF1Database.zip file:
SG247148\sampcode\wid\dbAccess\ClipsAndTacksF1Database.zip
- ▶ Select the ClipsAndTacksF1Database application and click *Finish* (Figure 10-44). The project appears under Other Projects.

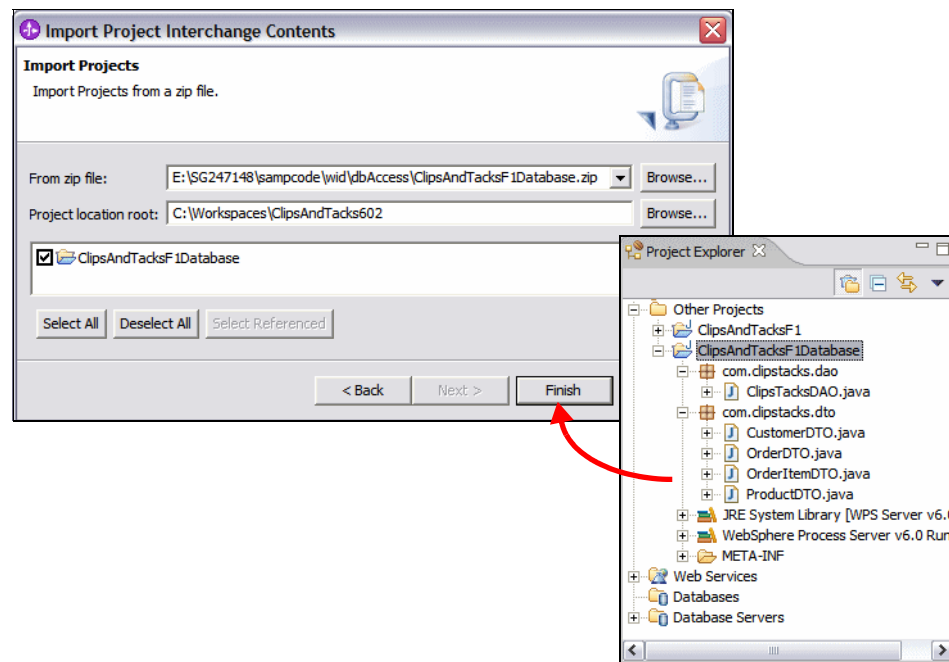


Figure 10-44 Import project containing database access code

Adding the Java project to the business process application

To attach the Java project to the business process application, follow these steps:

- ▶ In the Business Integration perspective select the `ClipsAndTacksF1` project and *Open Dependencies* (context menu).
- ▶ In the Dependencies editor, expand the Java section and click *Add*.
- ▶ Select the `ClipsAndTacksF1Database` project and click *OK*.
- ▶ Select the `ClipsAndTacksF1Database` project to verify that *Deploy with Module* is selected (Figure 10-45).

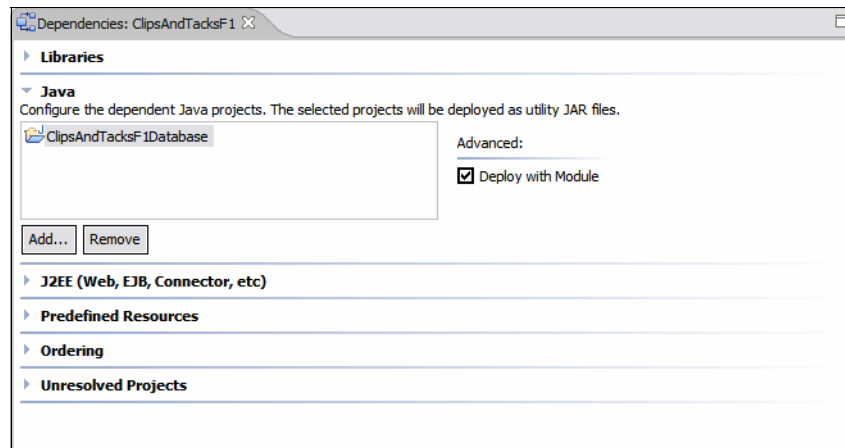


Figure 10-45 Dependencies Editor: Java section

Importing the Web application

We provide the `ClipsAndTacksF1Invoke` application as a WAR file that can be imported into Integration Developer:

- ▶ In the Web perspective, select *Dynamic Web Projects* and *Import* → *WAR file* (context menu).
- ▶ Click *Browse* to locate the WAR file:
samcode\wid\webfront\ClipsAndTacksF1Invoke.war
- ▶ Clear *Add module to an EAR project*.
- ▶ Click *Finish* (Figure 10-46).

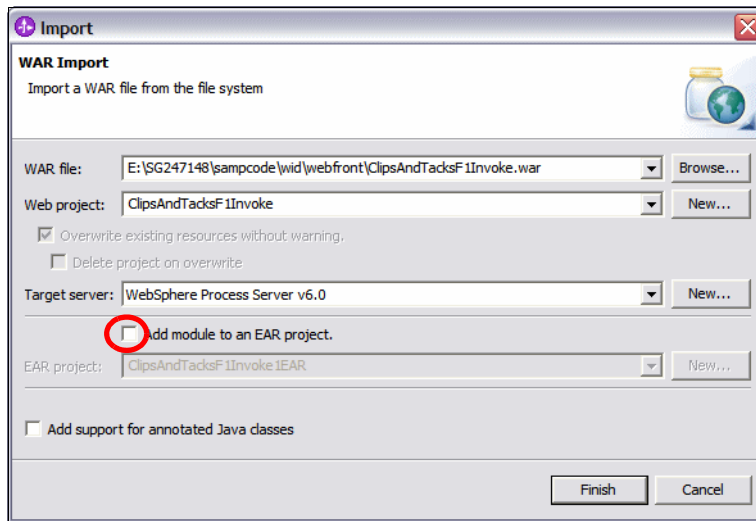


Figure 10-46 Import Web front-end WAR file

- ▶ We fix the errors resolving the references to the Java modules used for the database access by adding the `ClipsAndTacksF1Database` project to the build path of `ClipsAndTacksF1Invoke`:
 - In the Web perspective, select the `ClipsAndTacksF1Invoke` project and click *Properties* (context menu)
 - Select *Java Build Path* and then select the *Projects* tab.
 - Select the `ClipsAndTacksF1Database` project and then click *OK* (Figure 10-47).

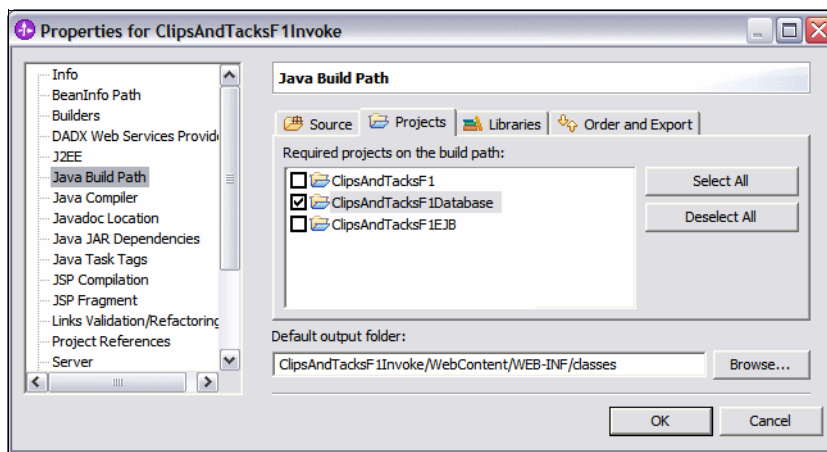


Figure 10-47 Add ClipsAndTacksF1Database to the Java build path

Data source reference

The data source reference is visible in the Web deployment descriptor as a resource reference with the name `ClipsAndTacks`. This reference has a WebSphere binding of `jdbc/cliptack`, therefore it refers to the data source defined in the server (Figure 10-48).

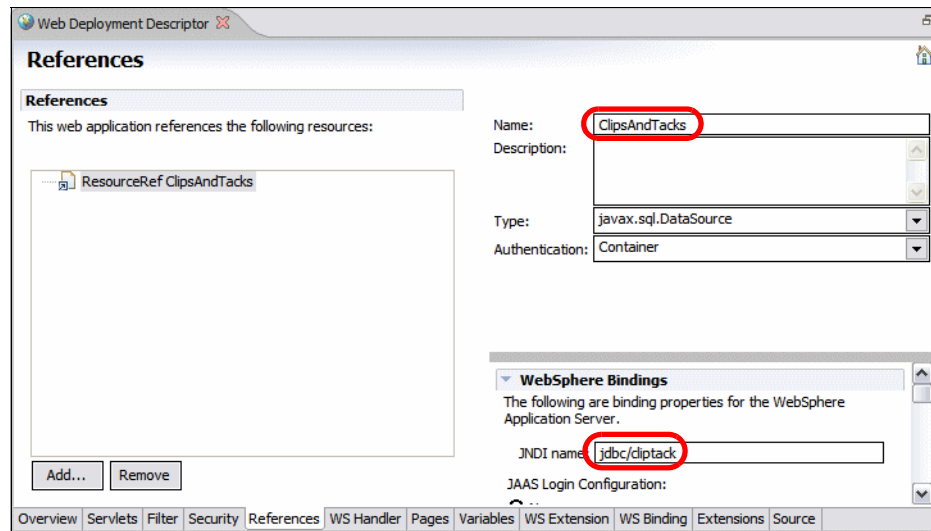


Figure 10-48 Web Deployment Descriptor data source resource reference

Attaching the Web front-end to the process application

The Web application invokes the process through the stand-alone reference. This is only possible if the Web application is part of the same enterprise application as the process.

To attach the Web application to the business process application, follow these steps:

- ▶ In the Business Integration perspective select the `ClipsAndTacksF1` project and *Open Dependencies* (context menu).
- ▶ In the Dependencies editor, expand the J2EE section and click *Add*.
- ▶ Select the `ClipsAndTacksF1Invoke` project and click *OK*.
- ▶ Select the `ClipsAndTacksF1Invoke` project to verify that *On Build Path* and *Deploy with Module* are both selected (Figure 10-49).

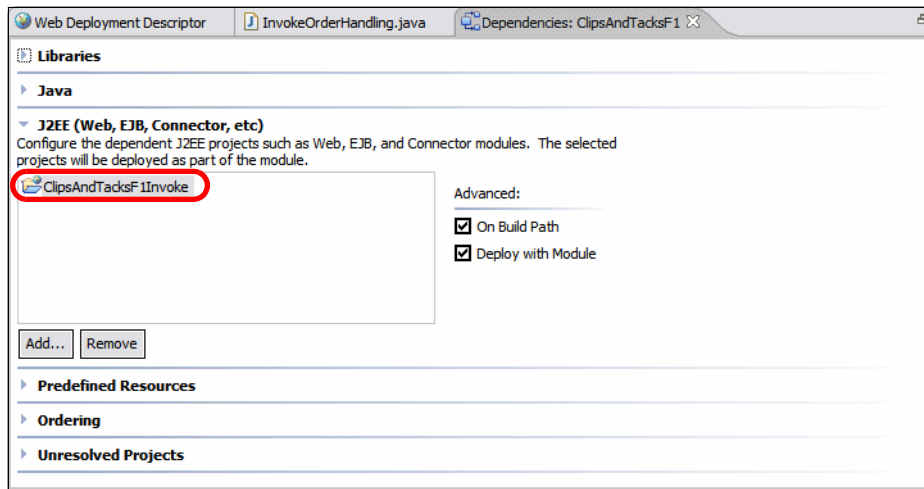


Figure 10-49 Dependencies editor to add Web project

- ▶ Save and close the Dependencies editor.

Tip: Adding a project using the Dependencies editor adds the project to the enterprise application. You can verify this by opening the deployment descriptor of the `ClipsAndTacksF1App` EAR. If you use the EAR deployment descriptor to add a Web project, the project might disappear when you deploy or redeploy the application. Always use the Dependencies editor! The Web application references the Java project as a JAR dependency.

Redeploy the application

To activate the Web front-end in the server, we have to redeploy the application:

- ▶ Select the server in the Servers view and *Publish* (context menu).

Initializing the database

The CLIPTACK database does exist at this point (see “Creating a data source for the database” on page 218), but the tables have not been defined.

To initialize the database, we run the `CreateDatabaseServlet`, which was imported as part of the `ClipsAndTacksF1Invoke` Web project:

- ▶ Expand the `ClipsAndTacksF1Invoke` project deployment descriptor.
- ▶ Expand *Servlets*, then select the `CreateDatabaseServlet` and *Run* → *Run on Server* (context).
- ▶ Select the Process Server, then click *Finish*.

- ▶ The servlet runs and displays its results:

```
Initializing database CLIPTACK
Connecting to database
Dropping existing tables...
drop table ct.orderitem
...
Creating tables...
Creating table CUSTOMER...
...
Creating foreign keys...
Inserting data into tables...
Insert table CUSTOMER...
...
End of database create: CLIPTACK
```

The data description language (DDL) statements and the sample data are retrieved from a properties file (`clipstacks.properties`). You can open the properties file from the Web perspective (*ClipsAndTacksF1Invoke* → *WebContent* → *WEB-INF* → *clipstacks.properties*).

- ▶ You can also use an external browser with the URL to initialize the order database:

<http://localhost:9080/ClipsAndTacksF1Invoke/CreateDatabaseServlet>

Tip: You can initialize the database at any time by rerunning the `CreateDatabaseServlet`.

Listing the database content

You can list the database at any time by running the `ListServlet`:

If you list the database during testing, you will notice that the order status does not yet get updated to either shipped or cancelled. This is addressed in “Implementing database update” on page 287.

Figure 10-50 shows the initial database content.

Customer data								
Number	Company	Contact	Address	Rating	Class	Credit	Pin	e-mail
12345	ABC Finance Ltd.	Marc Shankaran	1 Main Street, Buffalo, USA, 82840	777	GOLD	\$2000.00	12345	marc@abcfinance.com
11111	Auto Insurance Company	Richard Doe	27 New Street, Etobicoke, Canada, M8Z 2S6	666	SILVER	\$1300.00	11111	richard@autoinsurance.com
22222	ABC University	James James	8200 New Street, Markham, Canada, M9W 9M9	555	REGULAR	\$500.00	22222	james@abcuniversity.com

.

Customer Number

Product data			
Number	Name	Price	Description
RB-0001	All-In-One Printer	\$150.00	Marvellous printer, fax, copier
RB-0002	Manager Chair	\$79.00	Leather chair that turns and massages
RB-0003	5 MP Digital Camera	\$499.00	This digital camera takes MPEG4 movies
RB-0004	Cordless Phone with Answering Machine	\$89.00	This phone features a digital answering machine
RB-0005	3-Drawer File Cabinet	\$214.00	This cabinet has enough space for your lunch

Order data				
Number	Customer	Total Price	Status	Quantity/Product
3001	12345	\$308.00	SHIPPED	1 RB-0001
				2 RB-0002

Figure 10-50 Database content

Implementing customized JSPs for the BPC Explorer

At this point we can submit orders using the Web front-end, but before we do that, we want to improve the handling of the human tasks so that we do not have to enter all the data for the output message.

A customized human task JSP™ is invoked by the BPC Explorer instead of the standard JavaServer Faces pages. The customized JSPs can create the output message based on the input message and also provide actions. For example, we are able to approve or decline an order through a pull-down menu.

Importing the human task JSPs

We provide the human task JSPs in a Web application with the name `ClipsAndTacksF1BPC`. We have to import the WAR file and attach the Web project to the business process:

- ▶ Import the `ClipsAndTacksF1BPC.war` file from:
`SG247148\sampcode\wid\humantaskBPC\ClipsAndTacksF1BPC.war`
Be sure to clear *Add module to an EAR project*.
- ▶ Add `ClipsAndTacksF1Database` to the Java build path for `ClipsAndTacksF1BPC` (see “Importing the Web application” on page 252) to resolve the broken links to the database access module.
- ▶ In the Business Integration perspective, open the Dependencies editor for `ClipsAndTacksF1` (see “Attaching the Web front-end to the process application” on page 254).
- ▶ Expand the J2EE section and click *Add*. Select `ClipsAndTaskF1BPC` and click *OK* (Figure 10-51).

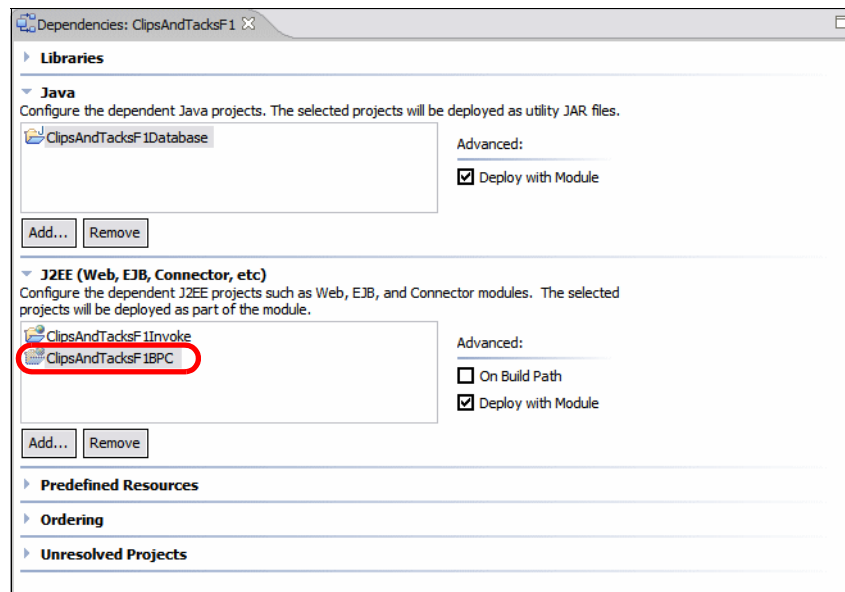


Figure 10-51 *ClipsAndTacksF1 Dependencies Editor*

- ▶ Redeploy the module. The module will be redeployed automatically if you selected *Enable Automatic Publishing* on the server (see “Starting the process server and administrative console” on page 216).

Human task JSPs

There are four JSPs in the `ClipsAndTacksF1BPC` project:

- ▶ `reviewOrderInput.jsp`—Handles the input message for order reviews
- ▶ `reviewOrderOutput.jsp`—Handles the output message for order reviews
- ▶ `shipOrderInput.jsp`—Handles the input message for order shipments
- ▶ `shipOrderOutput.jsp`—Handles the output message for order shipments

Data source reference

The `ClipsAndTacksF1BPC` Web project also contains the `ClipsAndTacks` resource reference for the data source. The human task JSPs have to maintain the database with the latest customer rating and available credit and update the order status as part of the `Review Order` human task.

Configuring the BPC Explorer for customized JSPs

The next step is to configure the BPC Explorer to invoke the customized JSPs.

- ▶ Open the `Review Order` activity from the assembly diagram or from the Business Integration view (under *Business Logic* → *Human Tasks*).
- ▶ Select the *Business Process Choreographer Explorer* under Client settings, and in the Properties view click *Add* to define a customized JSP.
- ▶ In the JSP Definition dialog select *Input message JSP* for type and *Potential owner* for apply to. Then click *Browse* and locate the `reviewOrderInput.jsp` in the `ClipsAndTacksF1BPC` project (Figure 10-52). Click *OK*.

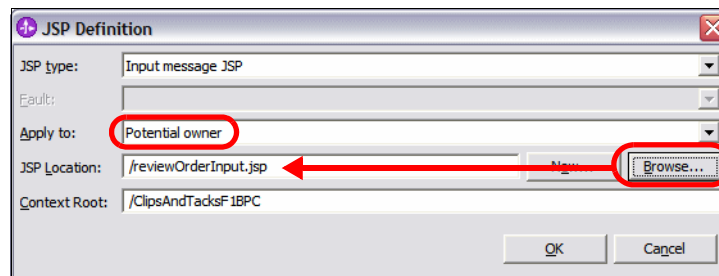



Figure 10-52 JSP selection for a human task

Note: At the time of writing, the following work-around was required to fix a problem populating the JSP Location field in Figure 10-52:

- ▶ Open the `.websettings` file in the `ClipsAndTacksF1BPC` project. You have to change the filter to display `.*` files. In the Project Explorer click the  icon and *Filters* and clear the `.*` entry.
- ▶ Remove the line `<lib-modules/>` from the `.websettings` file.
- ▶ Save and close the file.

- ▶ The JSP definition appears in the list.
- ▶ Click *Add* once more to define the *Output message JSP* as `reviewOrderOutput.jsp`.
- ▶ The list of JSPs for the Review Order activity is shown in Figure 10-53.

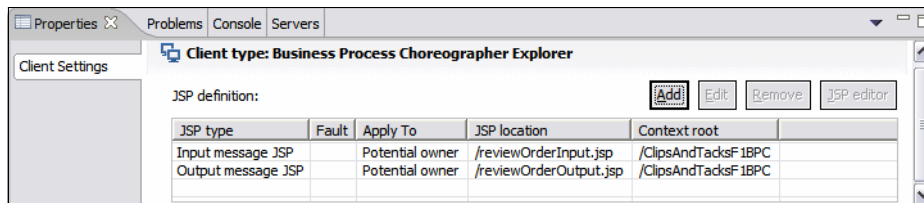


Figure 10-53 Human task JSPs for reviewing the order

Repeat this process for the Ship Order to Customer activity and define the `shipOrderInput.jsp` and `shipOrderOutput.jsp`.

Human task JSP processing logic

To understand how to implement a human task JSP, let us open the JSPs for the Review Order activity (`reviewOrderInput.jsp` and `reviewOrderOutput.jsp`).

The `reviewOrderInput.jsp` receives these objects in the request block:

- ▶ **message**—The data object of the input message.
- ▶ **messageMap**—A Java map holding XPath expression attribute names and their values of the input message. The XPath expression names mirror the layout of the data object. In our example, we do not have to retrieve the `messageMap`, as we already know the structure of the message data object.

The `reviewOrderOutput.jsp` receives the objects of the same names, but for the output message. **The output JSP has no access to the input message.**

In our case, the output message is basically the same as the input message, with a few fields changed (for example, the order status changes to APPROVED or DECLINED). To provide the output JSP with the input message, we pass the input message from the input JSP to the output JSP in the servlet request block.

Input message JSP

In this section we look at some code fragments of the `reviewOrderInput.jsp`:

- ▶ Retrieve the input message:

```
com.ibm.ws.bo.impl.BusObjImpl inputOrder =
    (com.ibm.ws.bo.impl.BusObjImpl)request.getAttribute("message");
```

- ▶ Pass the input message to the output JSP:

```
request.setAttribute("inputMessage", inputOrder);
```

- ▶ Retrieve the input data objects:

```
DataObject order    = inputOrder.getDataObject("Input");
DataObject customer = order.getDataObject("Customer");
DataObject procpref = order.getDataObject("ProcessingPreference");
java.util.List items = order.getList("OrderItems");
```

- ▶ Retrieve data items for display in the JSP:

```
int customerNumber = customer.getInt  ("CustomerNumber");
String company     = customer.getString("CompanyName");
double totalPrice  = order.getDouble  ("TotalPrice");
```

- ▶ Update the database with new customer rating and available credit (this happens if an order was automatically approved, but over the credit):

```
boolean automatic = procpref.getBoolean("automaticApproval");
if (automatic) {
    clipstacksDAO.updateCustomer(customer.getInt("CustomerNumber"),
        customer.getInt("Rating"),
        new java.math.BigDecimal(customer.getDouble("AvailableCredit")));
    orderStatus = "Auto-APPROVED, over available credit";
}
```

- ▶ The rest of the JSP displays the input message in an HTML table.

Output message JSP

In this section we look at some code fragments of the `reviewOrderOutput.jsp`:

- ▶ Retrieve the output message:

```
com.ibm.ws.bo.impl.BusObjImpl outputOrder =
    (com.ibm.ws.bo.impl.BusObjImpl)request.getAttribute("message");
```

- ▶ Retrieve the input message (passed from the input JSP):

```
com.ibm.ws.bo.impl.BusObjImpl inputOrder =
    com.ibm.ws.bo.impl.BusObjImpl request.getAttribute("inputMessage");
```

- ▶ When an output JSP is called from the **Work On** action in BPC Explorer, a prefix is set to be used in JSP fields for the output message:

```
String prefix = (String)request.getAttribute("prefix");
```

- ▶ Prepare the output message:

```
DataObject customerIn = orderIn.getDataObject("Customer");
java.util.List itemsIn = orderIn.getList("OrderItems");
order.setDataObject("Customer", customerIn);
order.setList      ("OrderItems", itemsIn);
order.setInt      ("OrderNumber", orderIn.getInt("OrderNumber") );
order.setDouble   ("TotalPrice", orderIn.getDouble("TotalPrice") );
order.setString   ("OrderStatus", orderIn.getString("OrderStatus") );
```

- ▶ Display a pull-down menu to approve or decline the order:

```
<h3>Approve or decline the order and click Complete</h3>
<SELECT id="status" name="{prefix}/Output/OrderStatus">
  <OPTION value="APPROVED" selected>Approve</OPTION>
  <OPTION value="DECLINED">Decline</OPTION>
</SELECT>
```

Note the prefix and the XPath expression for the name of the field:

```
{prefix}/Output/OrderStatus
```

This notation must be used to store values from JSP fields into the output message.

The processing for the **Ship Order to Customer** activity is very similar:

- ▶ The shipOrderInput.jsp also updates the database with the latest customer information and passes the input message to the output JSP.
- ▶ The shipOrderOutput.jsp creates the output message (product shipment).

Running the application in the test environment

The application is now ready to be run with the Web front-end and the customized human task JSPs.

Redeploy the application by removing it from the server and then adding it again if you did not select auto publish to server (select *Add and remove projects* in the Servers view).

Using the Web front-end

To start the Web front-end, select the ClipsAndTacksF1Invoke project and *Run* → *Run on Server* (context). Alternatively use an external browser with the URL:

`http://localhost:9080/ClipsAndTacksF1Invoke/`

- ▶ The Welcome (login) page is displayed (Figure 10-54).

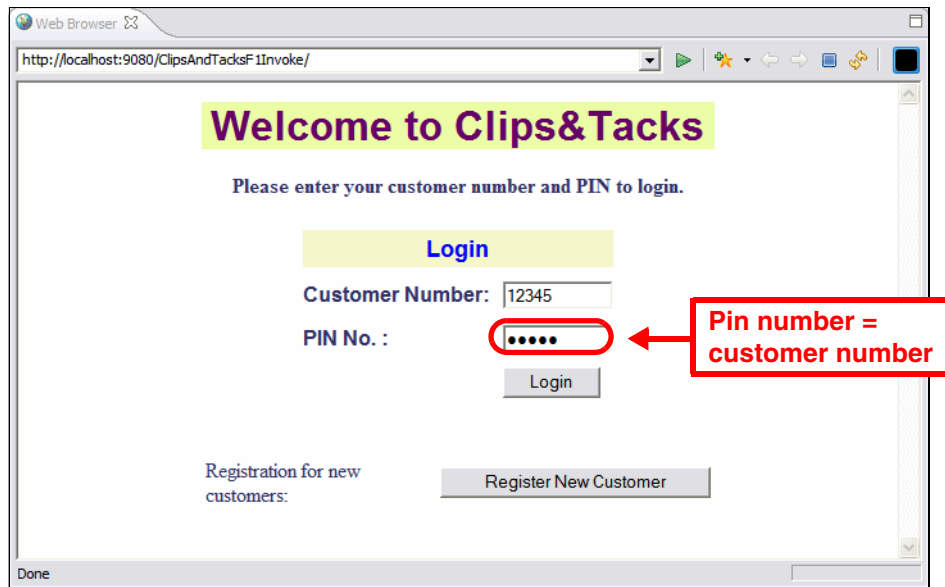


Figure 10-54 ClipsAndTacks: Welcome page

- ▶ Enter a customer number and pin and click *Login*.
- ▶ Place an order by selecting some products (Figure 10-55). Notice how the total amount is updated when you select products. This is done through a JavaScript™ routine.

Note that the total price is over \$750. Therefore, the order will go to the order manager for approval through a human task.

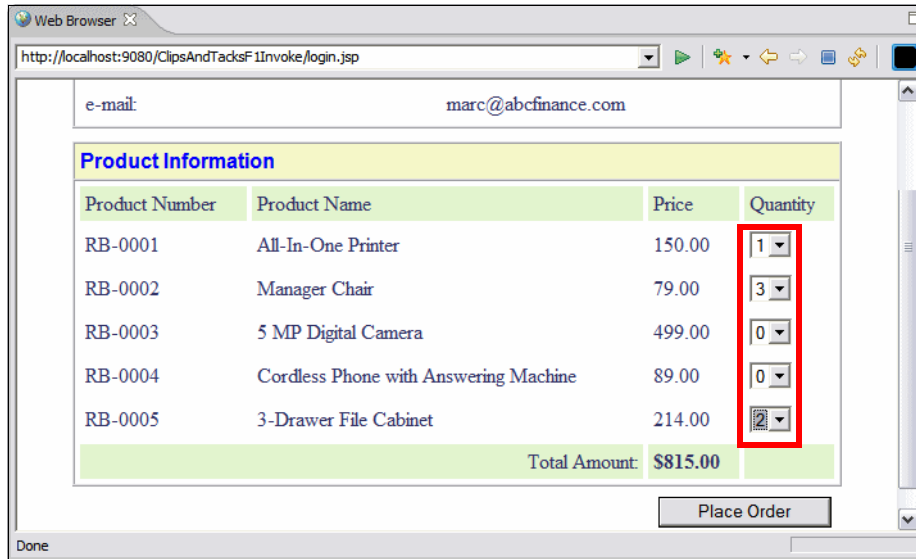


Figure 10-55 ClipsAndTacks: Placing an order

- ▶ Click *Place Order* and the order is placed and a confirmation is issued (Figure 10-56). The business process is invoked through the stand-alone reference.



Figure 10-56 ClipsAndTacks: Order confirmation

Using the human task JSPs with the BPC Explorer

The business process has been invoked and is waiting for human task interaction:

- ▶ Start the BPC Explorer (*Launch* → *Business Process Choreographer Explorer* in the Servers view) or enter the URL in a browser window:
`http://localhost:9080/bpc`
- ▶ The BPC Explorer opens with the My Tasks view and the Review Order activity is visible (Figure 10-57).
- ▶ You can click the task name to see its input and output messages, or you can select the task and click *Work on*.

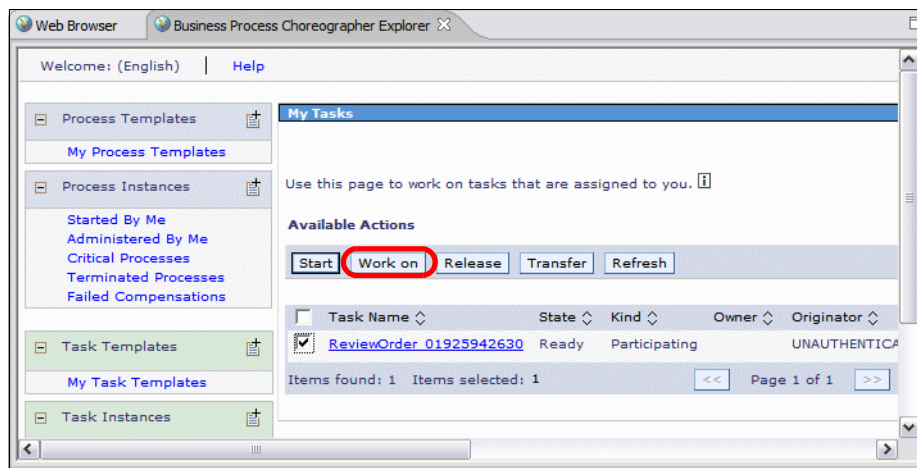


Figure 10-57 BPC Explorer with custom JSPs

- ▶ After clicking *Work on*, the custom JSP shows the input message and prompts to approve or decline the order (Figure 10-58).

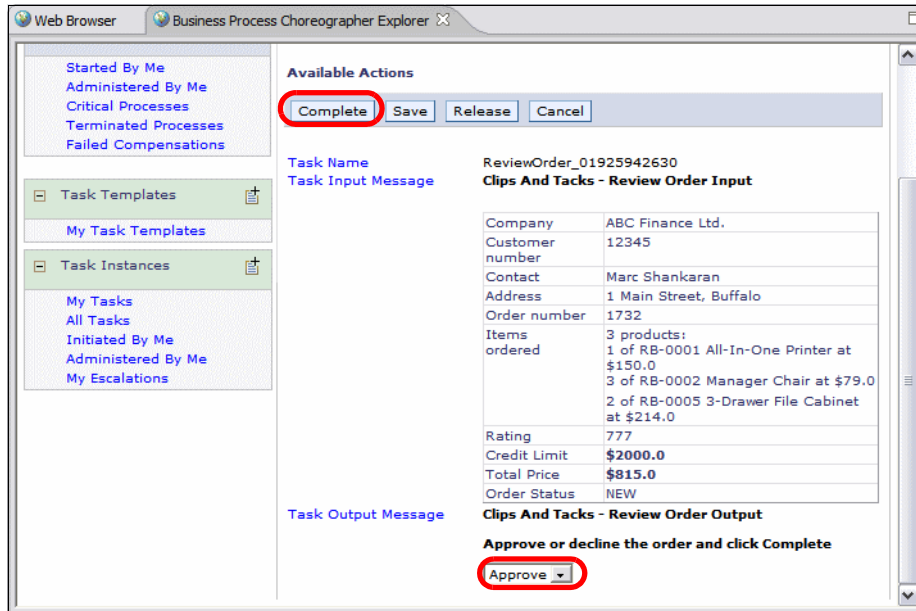


Figure 10-58 BPC Explorer: Custom JSPs for order review

- ▶ Select *Approve* and click *Complete*. After refreshing the task list, the process is now waiting for shipping. Select the *ShipOrdertoCustomer* task and click *Work on*. The custom JSPs for shipping are displayed (Figure 10-59).

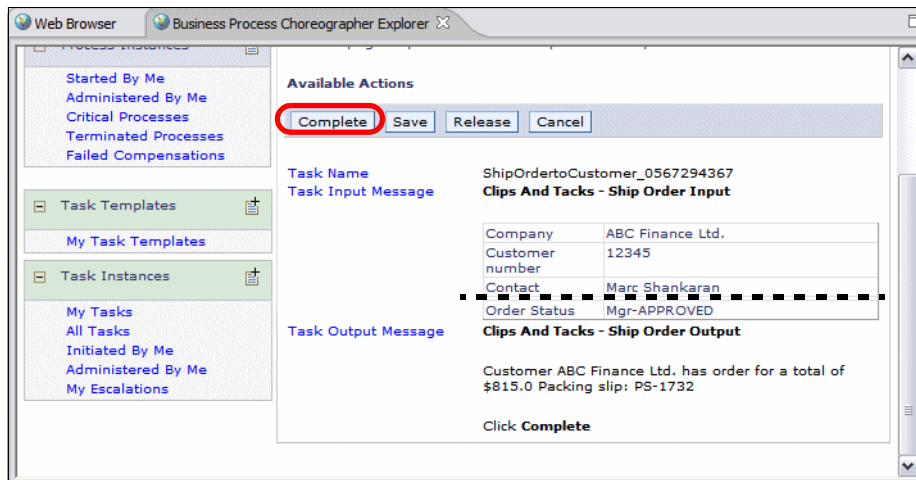


Figure 10-59 BPC Explorer: Custom JSPs for order shipment

- ▶ Submit another order for over \$750.00, but this time, decline the order.

Implementing an external Web service

To demonstrate additional functionality of the process server and business process assembly, we implement a Web service that is called by the Check Customer Account Status activity.

We already imported the `CreditRating` JavaBean that calls the Web service in “Implementing the Java activities” on page 233.

Importing the Web service

We do not describe how to create the Web service. If you want to learn about creating Web services, refer to the Redbooks publication, *WebSphere Version 6 Web Services Handbook Development and Deployment*, SG24-6461.

To import the Web service application, follow these steps:

- ▶ Select *File* → *Import* → *EAR File*.
- ▶ Click *Browse* and locate the file:

```
SG247148\sampcode\wid\webservice\CreditServiceEAR.ear
```

- ▶ Click *Finish*.

The Web service application consists of a Web project, `CreditServiceWeb`, which contains a JavaBean, `CreditBean`. This JavaBean has one method, `checkCredit`, that has been turned into a Web service:

```
public int checkCredit(CustomerBean customer) {
    int rating = 0;

    // calculate new rating between 500 and 800
    rating = 650 + randomGenerator.nextInt()%150;

    System.out.println("Web Service: Create rating for " +
        customer.getCustomerNumber() + " " +
        customer.getCompanyName() + " ==> rating: " +
        rating);

    return rating;
}
```

As you can see, we do not use a real external service; we only simulate such a response.

To invoke the Web service, the business process application requires the WSDL file from the service project:

```
CreditServiceWeb/WebContent/WEB-INF/wsd1/CreditBean.wsdl
```

Adding the Web service to the assembly diagram

To attach the Web service to the business process, follow these steps:

- ▶ Copy the WSDL file into the business process project:
From: CreditServiceWeb/WebContent/WEB-INF/wsd1/CreditBean.wsdl
To: ClipsAndTacksF1
- ▶ The Web service shows up in the Business Integration view under Web Service Ports. The CreditBean interface is also displayed (Figure 10-60).

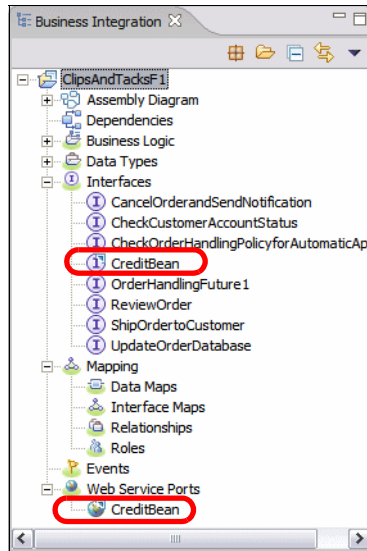


Figure 10-60 Business Integration view with Web service

- ▶ Open the assembly diagram.
- ▶ Select the CreditBean and drag it into the diagram, next to the Check Customer Account Status activity.
- ▶ In the Component Creation dialog, select *Import with Web Service Binding* (Figure 10-61). Click *OK*.

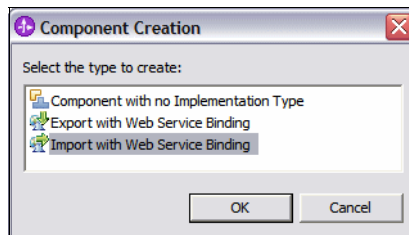


Figure 10-61 Component creation for a Web service

- ▶ Select the new component (CreditBeanImport1) and change the name to CheckCreditImport.
- ▶ Select the wire icon and draw a connection from Check Customer Account Status to Check Credit Import. When prompted:
 - Click *OK* to create a matching reference on the source node.
 - Click *No* to use the WSDL interface (instead of converting to a Java interface).
- ▶ You may see the following warning message in the Problems view:

The component implementation "processes/orderhandlingfuture1/CheckCustomerAccountStatus_1292162843" is missing one or more asynchronous response methods for the declared references. Select "Synchronize Interfaces and References...->to Implementation" action to correct it.
- ▶ To correct this select Check Customer Account Status in the Assembly Diagram and then select *Synchronize Interfaces and References* → *to Implementation* (context menu) as suggested.
- ▶ The complete assembly diagram is shown in Figure 10-62.

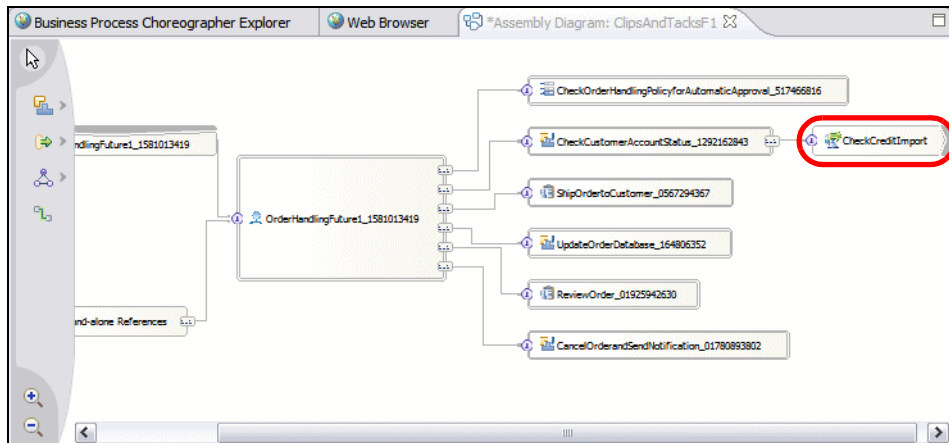


Figure 10-62 Assembly diagram with Web service invocation

Invoking the Web service from an activity

We already imported the CreditRating bean (see “Implementing the credit check” on page 234). Now we activate the call of the Web service:

- ▶ Open the CreditRating bean (in the ClipsAndTacksF1 project).
- ▶ Set the flag to invoke the Web service to true:

```
static boolean useWebservice = true;
```

- ▶ The code to invoke the Web service is shown here:

```
Service creditService =
    (Service)serviceManager.locateService("CreditBeanPartner");
// create data objects for Web Service call
DataObject wsInput = boFactory.createByElement(namespace,"checkCredit");
DataObject customerBean = boFactory.create(namespace, "CustomerBean");
customerBean.setInt("customerNumber", customerNumber);
customerBean.setString("companyName", companyName);
wsInput.setDataObject("customer", customerBean);
// invoke Web Service
System.out.println("...invoking checkCredit Web Service ...");
DataObject wsResult = (DataObject)creditService.
    invoke("checkCredit", wsInput);
// retrieve result from Web Service result
newRating = wsResult.getInt("checkCreditReturn");
```

Note: The code in the CreditRating bean resets the useWebservice flag to false if an error occurs in the Web service. This allows for continued testing without having to fix the application immediately.

Deploying the Web service

The Web service must be installed in a server. Add the CreditServiceEAR to the test server by selecting *Add and remove projects* in the Servers view.

Testing the application with the Web service

Now we can test the Web service invocation by submitting an order with a total price of less than \$750.00. The Check Customer Account Status activity is only invoked if the order is automatically approved.

The Console shows the trace of the Web service execution:

```
Check Customer Account Status invoked
Rating/Credit before: 12345: 777/2000.0
...invoking checkCredit Web Service ...
...
Web Service: Create rating for 12345 ABC Finance Ltd. ==> rating: 730
Rating/Credit after : 12345: 730/1995.3 (0.1)
```

After the order is shipped using the BPC Explorer, further messages in the Console show that the new rating and available credit are updated in the database, and the order is shipped:

```
Customer updated: 12345 (730,1995.299999999999)
Shipment PS-1580 for customer ABC Finance Ltd. status is: SHIPPED
Update Order Database invoked
```


Changing the port for the Web service

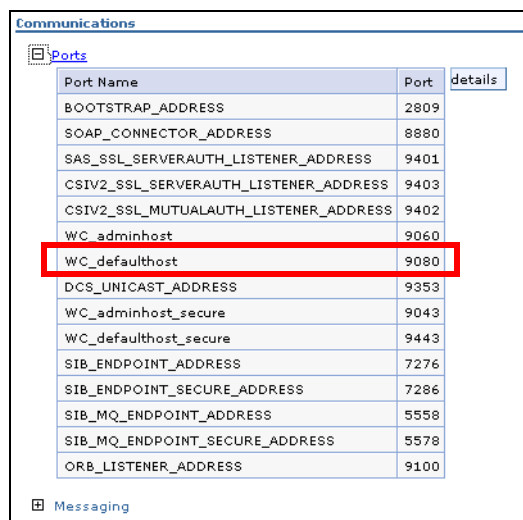
If you have more than one server defined, your port numbers might be different. The first server you define is assigned port 9080 for HTTP traffic, the second server 9081 and so forth. This is to avoid port number conflicts when running more than one server at the same time.

Note: You only have to change the port numbers manually if you *imported* the solution from the sample code and your server is *not* running on port 9080.

If you built the application yourself, or if your server is using port 9080, you do not have to change anything.

You can determine the port number of your server in the administrative console.

- ▶ Select *Servers* → *Application servers* → *server1*.
- ▶ In the Communications section, expand *Ports*.
- ▶ Verify the port setting for *WC_defaulthost* (Figure 10-63).



Port Name	Port	details
BOOTSTRAP_ADDRESS	2809	
SOAP_CONNECTOR_ADDRESS	8880	
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9401	
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9403	
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9402	
WC_adminhost	9060	
WC_defaulthost	9080	
DCS_UNICAST_ADDRESS	9353	
WC_adminhost_secure	9043	
WC_defaulthost_secure	9443	
SIB_ENDPOINT_ADDRESS	7276	
SIB_ENDPOINT_SECURE_ADDRESS	7286	
SIB_MQ_ENDPOINT_ADDRESS	5558	
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5578	
ORB_LISTENER_ADDRESS	9100	

Figure 10-63 Changing ports: Determining the server port number

If the port setting is not 9080, follow these steps:

- ▶ Change the port setting on the CheckCreditImport:
 - Open the assembly diagram.
 - Select CheckCreditImport.

- In the Properties view, select the *Bindings* tab and update the port (Figure 10-64).

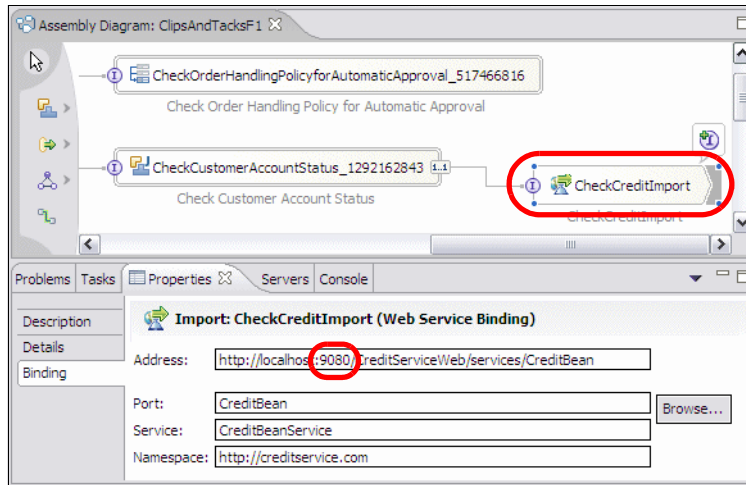


Figure 10-64 Change the Web service port on the Import binding

- ▶ Change the port setting in the WSDL file:
 - Open the CreditBean WSDL in the WSDL editor.
 - Change the port from 9080 to your port (Figure 10-65).

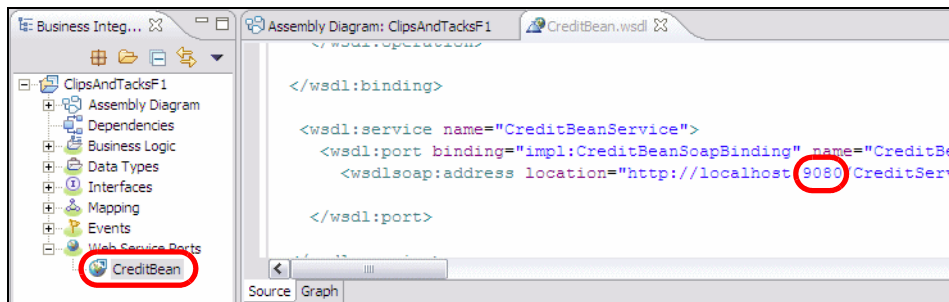


Figure 10-65 Changing the Web service port in the WSDL file

- ▶ Save the changes.

Generating a human task application

Integration Developer provides a feature for generating a user interface for human tasks, business processes, or a module—a new feature provided in 6.0.2 that minimizes the turn-out of the process to display a simple and neat user interface.

The generated user interface is targeting the human tasks—whether standalone human tasks or inline human tasks—inside a process.

Although it might be limiting the user in options for creating a fully customized user interface, it is sufficient for our testing purposes.

For the process in hand, because we do not have inline human tasks, we generate the user interface only for the standalone human tasks.

For generating the user interface, follow these steps:

- ▶ Select one of the human tasks that are available in the project tree view and *Generate User Interfaces* (Figure 10-66).

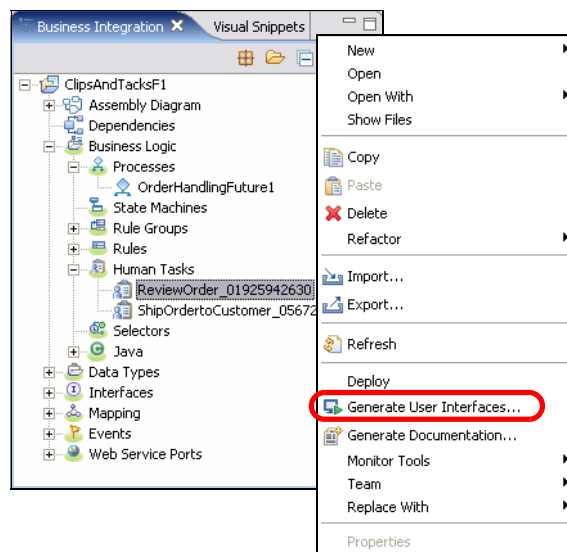


Figure 10-66 Generating a human task user interface

- ▶ A list of the available human tasks is displayed.
- ▶ Select both processes that are listed (Figure 10-67).



Figure 10-67 Selecting the humans tasks for the user interface

- ▶ In the JSF client configuration page (Figure 10-68), select a name for the dynamic Web project that will be created for the user interface, for example:
 ClipsAndTacksF1HumanUI
- ▶ A company logo can be provided as well to be part of the user interface; click *Browse* to select the logo:
 SG247148\sampcode\image\ClipsAndTacksImage.gif
- ▶ A set of styles are provided by default, select one of them to be the general theme of the user interface—here IBM Style has been selected.

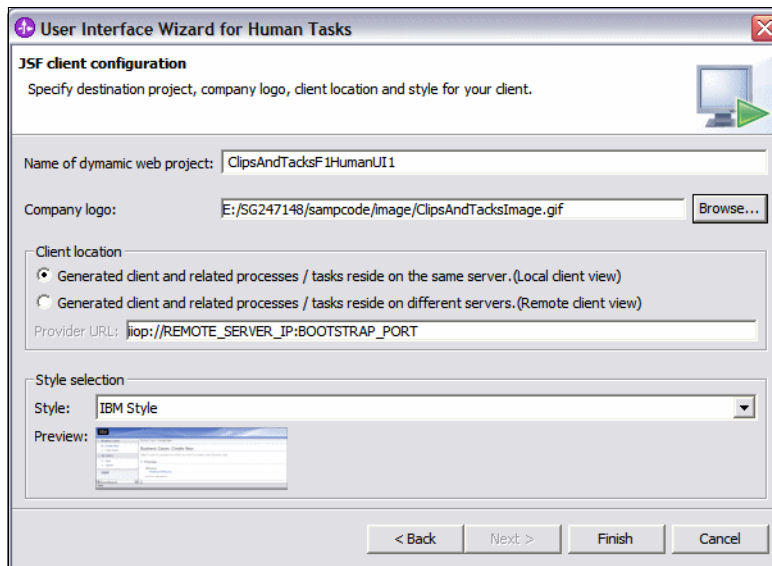


Figure 10-68 Generated user interface Web application

- ▶ The user interface is generated, and pop-up window displays the instructions for deploying the generated user interface (Figure 10-69).

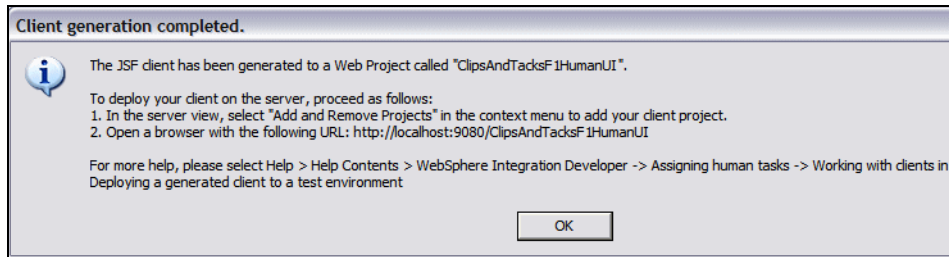


Figure 10-69 User interface has been created

After the user interface is generated and deployed to the server, we can start a process and test the user interface:

- ▶ Submit an order with an amount over \$750 (see “Using the Web front-end” on page 263).
- ▶ Open the URL that was provided at the end of the generation of the user interface, or select the ClipsAndTacksHumanUI project and *Run* → *Run on Server*.
- ▶ The Business User Client home page is displayed (Figure 10-70).

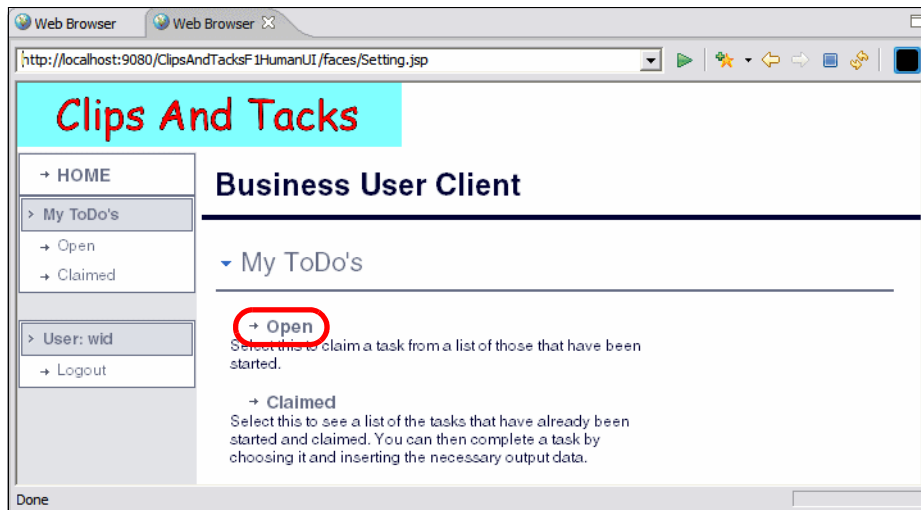


Figure 10-70 Generated user interface: Home page

- ▶ Click *Open* and the list of work items that are assigned to this user is displayed.
- ▶ Select the task that is required to be performed (Figure 10-71).

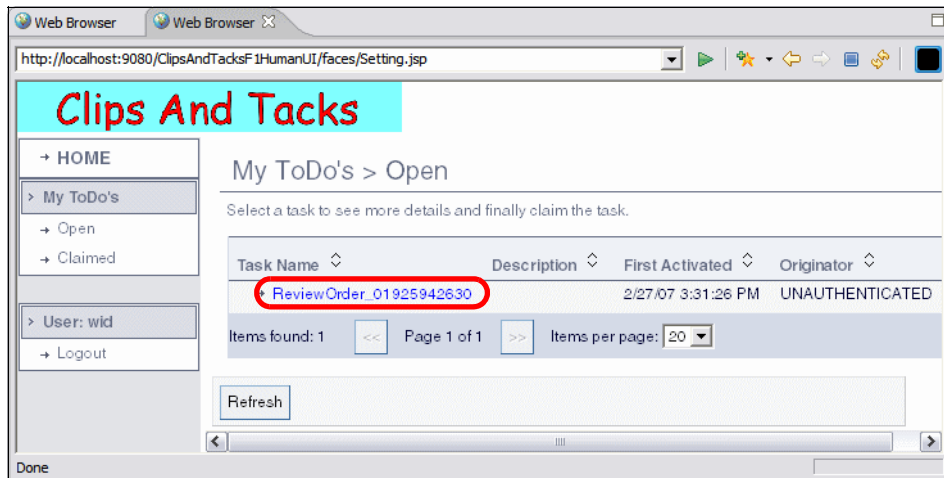


Figure 10-71 Selecting a task to perform

- The task is displayed with its input data (Figure 10-72).

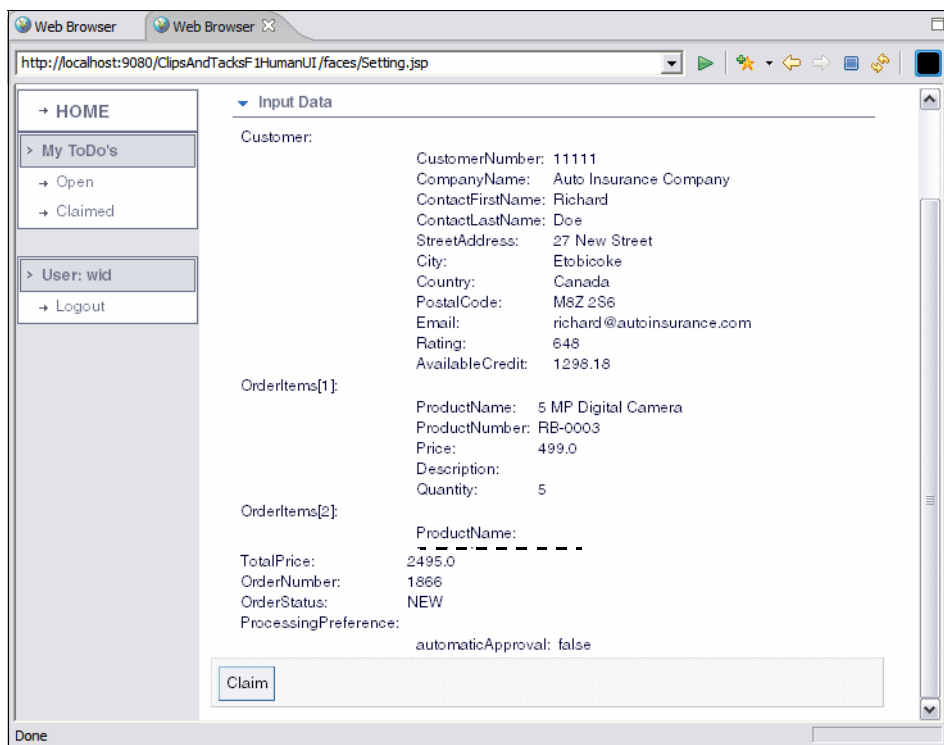


Figure 10-72 Claiming a task

- ▶ Note that by default two items of the OrderItems array are displayed. To display the actual number of items requires changing the generated JSPs.
- ▶ Click *Claim*.
- ▶ The list of output fields to be filled with data is displayed. This is similar to the BPC Explorer without tailored JSPs that require the output messages to be filled by hand.
- ▶ Fill the output data and click *Complete* to finish the task. Note that all fields must be filled with data or you get validation errors. Enter APPROVED for the order status.
- ▶ Refresh the list and the ship order task is displayed. Perform the same operations as for the review order task.

Improving the generated user interface

Improving the generated user interface is a complicated task. Here is a list of improvements that can be made to the input and output JSPs:

```
ToDo_input1_task_ShipOrdertoCustomer_0567294367.jsp
ToDo_output1_task_ShipOrdertoCustomer_0567294367.jsp
ToDo_input2_task_ReviewOrder_01925942630.jsp
ToDo_output2_task_ReviewOrder_01925942630.jsp
```

- ▶ Display up to five order items, dependent on input data:

```
<jsp:useBean id="todoMessageHandler" scope="session"
             class="com.ibm.wbit.tel.client.jsf.handler.TODOMessageHandler" />
com.ibm.wbit.tel.client.jsf.bean.TODOInstance instance =
             todoMessageHandler.getToDoInstance();
value = (String)instance.getInputValues()
             .get("/Input/OrderItems[1]/ProductName");
if ( value != null ) {
    // display oirder item 1
}
```

- ▶ Remove fields that are not required (product description).
- ▶ Make the output fields that cannot be changed as read-only:

```
<h:inputText id="OutputOrderItems1ProductName_ID"
             styleClass="ViewEntryField"
             value="#{todoMessageHandler.todoInstance.outputValues
             ['/Output/OrderItems[1]/ProductName']}" required="true"
             readonly="true"/>
```

- ▶ Possibly remove the validation for optional fields.

- ▶ Change the output order status field to a drop-down list for selection of APPROVED or DECLINED:

```
<h:selectOneMenu styleClass="selectOneMenu" id="menu1"
    value="#{ToDoMessageHandler.todoInstance.outputValues
        ['/Output/OrderStatus']}">
    <f:selectItem itemValue="APPROVED" itemLabel="Approve" />
    <f:selectItem itemValue="DECLINED" itemLabel="Decline" />
</h:selectOneMenu>
```

- ▶ Rearrange fields into a better sequence, for example, move order number and total price to the top for the shipping output JSP.

The most needed improvement is to copy the input data to the output data where appropriate, so that the user only has to enter the new or changed data. This can be done in the JavaBean `com.ibm.wbit.tel.client.jsf.bean.ToDoInstance` in the `setOutputValues` method:

```
public void setOutputValues(HashMap outputValues) {
    String type = getOutputMessageType();
    //System.out.println("set output values:" + type);
    this.outputValues = outputValues;
    java.util.Set keys = inputValues.keySet();
    java.util.Iterator it = keys.iterator();
    while (it.hasNext()) {
        String key = (String)it.next();
        Object value = inputValues.get(key);
        //System.out.println("key=" + key + " value="+ value);
        if (type.indexOf("Ship") > 0 ) { // ShipOrder
            if ( key.indexOf("Rating") > 0 ) continue;
            if ( key.indexOf("AvailableCredit") > 0 ) continue;
            if ( key.indexOf("Processing") > 0 ) continue;
            if ( key.indexOf("Status") > 0 ) continue;
            if ( key.equals("/Input/OrderNumber") )
                this.outputValues.put
                    ("/Output/PackingSlipNumber", "PS-" + value);
        }
        this.outputValues.put("/Output" + key.substring(6), value);
    }
}
```

This method copies the input data hash table to the output data hash table. The hash table keys are in the format:

```
/Input/Customer/ContactFirstName
/Input/OrderNumber
```

The corresponding output hash table keys are:

```
/Output/Customer/ContactFirstName
/Output/OrderNumber
```


For the review human task the output is the same business object (Order), but for the shipping human task the output object (ProductShipment) is different and some fields are omitted (OrderStatus, ProcessingPreference, Rating) and one field is new (PackingSlipNumber).

The finished tailored generated human task applications is provided in:

SG247148\sampcode\wid\humanGeneratedUI\ClipsAndTacksF1HumanGen.zip

Implementing a customized human task application

The BPC Explorer provides the functionality to work with human tasks and, with the addition of custom JSPs, it is reasonably functional.

Human tasks can also be processed using an API that allows us to search for human tasks, claim a task for processing, and complete or release a task. Using this API we can build a human task application that is independent of the BPC Explorer.

Importing the human task application

We provide this application in an interchange file:

SG247148\sampcode\wid\humantaskApp\ClipsAndTacksFxHumanApp.zip

Note: This interchange file includes the security code described in “Implementing security for human tasks” on page 304. The application with security also runs in a server where security is not enabled. This application also includes the code necessary to handle the Future 2 application that we develop in Chapter 16, “Implementing the Future 2 process using WebSphere Integration Developer” on page 509.

To import the application, follow these steps:

- ▶ Select *File* → *Import* → *Project Interchange*.
- ▶ Click *Browse* to locate the ClipsAndTacksFxHumanApp.zip file.
- ▶ Select the projects, ClipsAndTacksFxHumanCustomEAR and ClipsAndTacksFxHumanCustomWeb. We have already have the ClipsAndTacksF1Database project in our workspace so do not select that project. Click *Finish*.

Add the ClipsAndTacksHumanFxCustomEAR to the server using *Add and remove projects*.

Architecture

The human task application consists of several JavaServer Faces JSPs and one JavaBean, **TaskQuery**, which performs all the interaction with the human task manager using the API.

The DTOs and the DAO in the ClipsAndTasksF1Database project are used for order processing and database access. ClipsAndTasksF1HumanCustomerWeb contains one extra DTO, TaskOrderDTO, which is a copy of the OrderDTO with two properties, taskId and taskState, added. The taskId is the task ID that is used to claim and complete a task. The taskState is either ready or claimed (there are other states that we do not handle).

The application must have access to the XSD file that contains the business objects and the associated namespace. This data is required when creating the output object of the task. The BusinessItems.xsd files are copied from the business process applications.

TaskQuery JavaBean

The TaskQuery bean performs all the processing through these methods:

- ▶ **getTaskMgr**—Retrieve the task manager session EJB at initialization time using a local EJB reference:

```
LocalHumanTaskManagerHome taskHome =
    (LocalHumanTaskManagerHome)initialContext.lookup
    ("java:comp/env/ejb/LocalHumanTaskManagerHome");
// Access the local interface of the local session bean
taskMgr = taskHome.create();
```

The local EJB reference (ejb/LocalHumanTaskManagerHome) is defined in the deployment descriptor. It points to the JNDI name:

```
com/ibm/task/api/HumanTaskManagerHome
```

- ▶ **execute**—Search for human tasks that are ready for processing. This method accepts a parameter (task name) that enables us to search for tasks of a certain name, for example, ReviewOrder.

The search is performed using SQL:

```
selectClause = "DISTINCT TASK.TKIID, TASK.NAME";
whereClause = "TASK.STATE = (TASK.STATE = TASK.STATE.STATE_READY OR
    TASK.STATE = TASK.STATE.STATE_CLAIMED) AND (TASK.KIND =
    TASK.KIND.KIND_PARTICIPATING OR TASK.KIND = TASK.KIND.KIND_HUMAN)";
orderClause = null;
threshold = null;
QueryResultSet result = taskMgr.query(selectClause,
    whereClause + " AND TASK.NAME LIKE '" + selectedTaskname + "%'",
    orderClause, threshold, null);
```

The task ID and the input data object can be retrieved from the result set:

```
result.next();
TKIID tkiid = (TKIID)result.getOID(1);
String taskname = result.getString(2);
ClientObjectWrapper input = taskMgr.getInputMessage(tkiid);
inputDO = (DataObject)input.getObject();
DataObject order = inputDO.getDataObject("Input");
.....
```

From the input data objects an array of TaskOrderDTO is built and is available for retrieval.

- ▶ **getOrders**—A JSP can use the `getOrders` method to retrieve the list of orders that require an action.
- ▶ **claimReviewComplete**—This method claims a review order task, performs the approve or decline action (parameter), updates the database, constructs the output message, and completes the task using the API:

```
// claim the task for processing
ClientObjectWrapper input = taskMgr.claim(tkiid);
// retrieve input data object
DataObject inputDO = (DataObject)input.getObject();
DataObject orderIn = inputDO.getDataObject("Input");
.....
// create output data object
ClientObjectWrapper output = taskMgr.createOutputMessage(tkiid);
DataObject outputDO = (DataObject)output.getObject();
DataObject orderOut = outputDO.getDataObject("Output");
if (orderOut == null) {
    ServiceManager serviceManager = new ServiceManager();
    BOFactory boFactory = (BOFactory)serviceManager
        .locateService("com/ibm/websphere/bo/BOFactory");
    orderOut = boFactory.create(namespace, "Order");
    outputDO.setDataObject("Output", orderOut);
}
.....
// complete the task
taskMgr.complete(tkiid, output);
```

- ▶ **claimShipComplete**—This method is similar to the `claimReviewComplete` method to perform the ship order operation.
- ▶ **claimOrder**—This method claims a task but does not complete it. We added this method so that the order manager can claim the task and complete it later. This should allow for a correct timing of the task in the Monitor.
- ▶ **getOrder**—This method retrieves the last order that was processed (and saved) by any of the `claim` methods.
- ▶ **setFutureVersion**—Store the version of the application.

JavaServer Faces pages

The JSF pages are:

- ▶ **index.jsp**—Home page to select the task to be performed (review or ship).
- ▶ **myReviews.jsp**—Retrieve the orders to be reviewed using the TaskQuery bean. Display the orders in a table, with a pull-down menu for the action (approve or decline) and a push button (Process) to complete the task.
- ▶ **myShipments.jsp**—Similar to myReviews.jsp for shipped orders. A shipped order is sent to the packingSlip.jsp to print the packing slip.
- ▶ **packingSlip.jsp**—Displays the packing slip for a shipped order.

Additional pages are used for the home page and for security:

- ▶ **index.html**—A simple home page that forwards to the index.jsp.
- ▶ **login.jsp** and **loginError.html**—These pages are used for login of a user (for example, order manager) when running with security (see “Implementing security for human tasks” on page 304).

Here are some extracts of the JSP code for review order:

- ▶ Access the TaskQuery bean on the HTTP session and call the execute method:

```
com.clipstacks.human.TaskQuery taskQuery =
    (TaskQuery)session.getAttribute("clipsTacksQuery");
if (taskQuery == null) {
    taskQuery = new com.clipstacks.human.TaskQuery();
    session.setAttribute("clipsTacksQuery", taskQuery);
}
taskQuery.execute("ReviewOrder");
```

- ▶ Access the number of orders and return if no orders are found:

```
if (taskQuery.getOrders().length == 0) { // no orders found
    FacesContext facesContext = FacesContext.getCurrentInstance();
    facesContext.getExternalContext().redirect
        ("faces/index.jsp?message=No orders for review found");
}
```

- ▶ The orders are displayed in a JSF data table.
- ▶ A hidden field is used to pass the order number when submitting an order for processing:

```
<h:inputHidden id="reviewNumber"
    value="#{varOrders.orderNumber}"></h:inputHidden>
```

- ▶ A pull-down menu is used for the action (approve, decline, or claim):

```
<h:selectOneMenu styleClass="selectOneMenu"
                id="orderAction">
    <f:selectItem itemValue="NONE" itemLabel="Select an action" />
    <f:selectItem itemValue="APPROVED" itemLabel="Approve" />
    <f:selectItem itemValue="DECLINED" itemLabel="Decline" />
    <f:selectItem itemValue="CLAIMED" itemLabel="Claim" />
</h:selectOneMenu>
```

- ▶ The process button is a JSF command button:

```
<hx:commandExButton type="submit" value="Process"
                    styleClass="commandExButton" id="completeButton"
                    action="#{pc_MyReviews.doCompleteButtonAction}">
</hx:commandExButton>
```

- ▶ The process button invokes the action logic (doCompleteButtonAction) in the JSF Java class (MyReviews.java):

```
// get the action (approve or decline)
String action = (String)getOrderAction().getValue();
// get the order number from the hidden field
int reviewNumber = ((Integer)getReviewNumber().getValue()).intValue();
System.out.println("Action="+action + " order "+reviewNumber);
String message = null;
// verify that an action was selected
if (!action.equals("APPROVED") && !action.equals("DECLINED") &&
    !action.equals("CLAIMED")) {
    message = "No action (Approve/Decline) was selected";
} else {
    // get the TaskQuery bean and process the order
    TaskQuery taskQuery = (TaskQuery)getSessionScope()
                          .get("clipsTacksQuery");

    if (taskQuery == null) {
        message = "ReviewOrder: Session data not available";
    } else {
        if (action.equals("CLAIMED"))
            message = taskQuery.claimOrder(reviewNumber);
        else // APPROVED or DECLINED
            message = taskQuery.claimReviewComplete(action, reviewNumber);
    }
}
// prepare the response message
getRequestScope().put("message", message);
return null;
```

- ▶ Shipped orders are forwarded to the packingSlip JSP (doShipButtonAction in MyShipments.java):

```
getRequestScope().put("order", taskQuery.getOrder());
```

Using the human task application

To see how the human task application works, submit a few orders using the Web front-end. Then start the human task application by selecting the Web project (ClipsAndTacksFxHumanCustomWeb) and *Run* → *Run on Server* (context menu):

- ▶ The human task application starts with the home page (Figure 10-73).

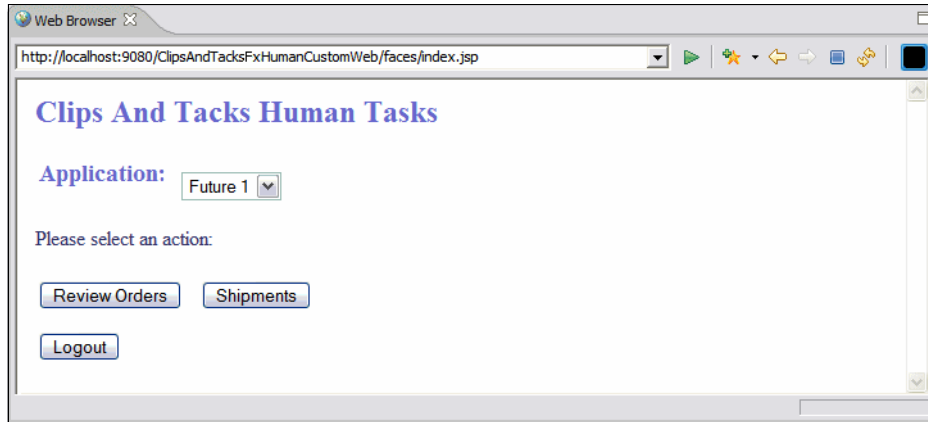


Figure 10-73 Human task application: Home

- ▶ Click *Review Orders* and the orders to be reviewed are displayed (Figure 10-74).

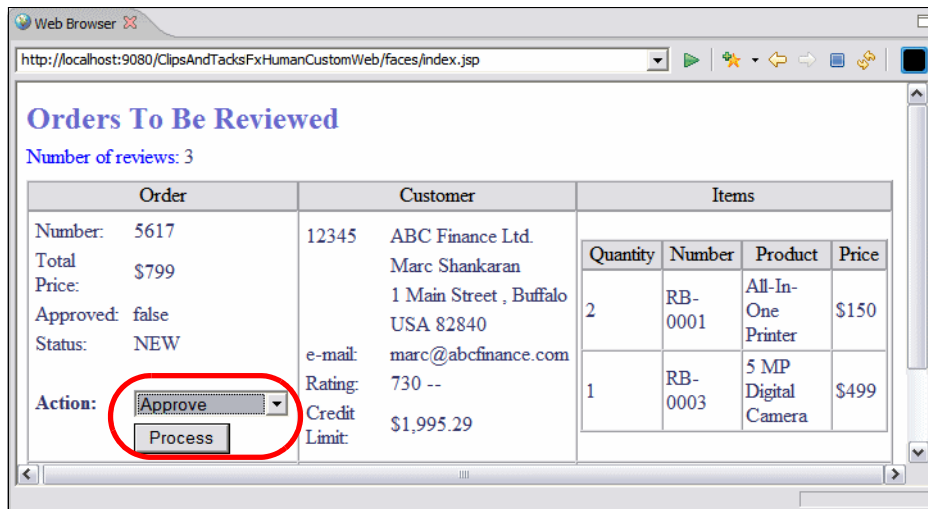


Figure 10-74 Human task application: Orders for review (1)

- ▶ Pick an order, select the action, and click *Process*.
 - **Approve**—The order is claimed and approved and proceeds to shipping.
 - **Decline**—The order is claimed and declined and proceeds to cancellations.
 - **Claim**—The order is claimed and can be approved or declined later. The order stays in the list with the text Task is CLAIMED.

Note: We implemented the claim action so that we can simulate the order manager activity of *thinking* about whether an order should be approved or declined. This action also locks the task from any other user.

- ▶ The page is redisplayed with the message and the order disappears for approve and decline actions (Figure 10-75).

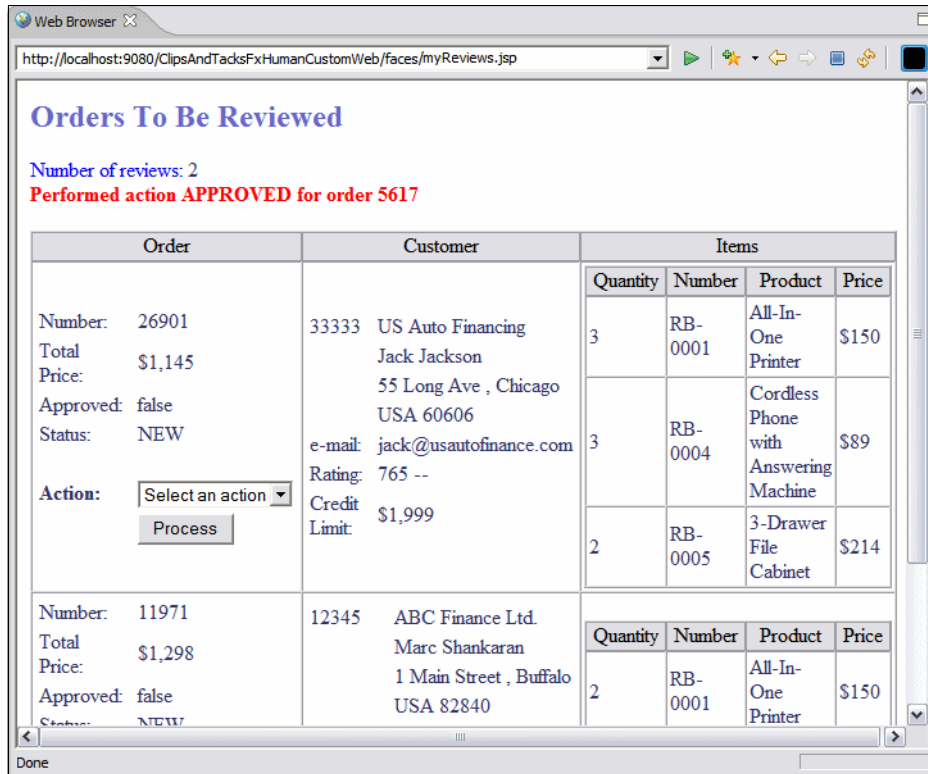


Figure 10-75 Human task application: Orders for review (2)

- ▶ Approve another order, then decline the remaining order and click *Process*. All orders to be reviewed are processed. Click *Home*, then click *Shipments*. There should be a few orders to be shipped (Figure 10-76).

The screenshot shows a web browser window with the URL `http://localhost:9080/ClipsAndTasksFxHumanCustomWeb/faces/index.jsp`. The page title is "Orders To Be Shipped" and it indicates "Number of shipments: 3".

Order	Customer	Items															
Number: 20812 Total Price: \$293 Approved: true Status: NEW Action: <input type="button" value="Claim"/> <input type="button" value="Complete"/>	22222 ABC University James James 8200 New Street , Markham Canada M9W 9M9 e-mail: james@abcuniversity.com Rating: 756 -- Credit Limit: \$520.1	<table border="1"> <thead> <tr> <th>Quantity</th> <th>Number</th> <th>Product</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>RB-0002</td> <td>Manager Chair</td> <td>\$79</td> </tr> <tr> <td>1</td> <td>RB-0005</td> <td>3-Drawer File Cabinet</td> <td>\$214</td> </tr> </tbody> </table>	Quantity	Number	Product	Price	1	RB-0002	Manager Chair	\$79	1	RB-0005	3-Drawer File Cabinet	\$214			
Quantity	Number	Product	Price														
1	RB-0002	Manager Chair	\$79														
1	RB-0005	3-Drawer File Cabinet	\$214														
Number: 5617 Total Price: \$799 Approved: false Status: APPROVED Action: <input type="button" value="Claim"/> <input type="button" value="Complete"/>	12345 ABC Finance Ltd. Marc Shankaran 1 Main Street , Buffalo USA 82840 e-mail: marc@abcfinance.com Rating: 730 -- Credit Limit: \$1,995.29	<table border="1"> <thead> <tr> <th>Quantity</th> <th>Number</th> <th>Product</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>RB-0001</td> <td>All-In-One Printer</td> <td>\$150</td> </tr> <tr> <td>1</td> <td>RB-0003</td> <td>5 MP Digital Camera</td> <td>\$499</td> </tr> </tbody> </table>	Quantity	Number	Product	Price	2	RB-0001	All-In-One Printer	\$150	1	RB-0003	5 MP Digital Camera	\$499			
Quantity	Number	Product	Price														
2	RB-0001	All-In-One Printer	\$150														
1	RB-0003	5 MP Digital Camera	\$499														

Figure 10-76 Human task application: Orders for shipment

- ▶ Select an order and click *Claim*, then think for 15 minutes.....
- ▶ Select an order and click *Complete*. The shipment is confirmed and the packing slip is displayed for printing (Figure 10-77).



Figure 10-77 Human task application: Packing slip for shipped order

Implementing database update

We have to update the order database with the order status when we cancel or ship an order. We will show two techniques for database update.

Using an information service

We have to update the order database when an order is shipped to reflect the change in the status of the order. We can do this by changing the Update Order Database component into an information service. Currently the Update Order Database activity is implemented in Java. However, it does not contain any functionality, it just prints a message to the console when it is invoked.

Using an information service activity allows us to directly access a relational database management system (for example, Cloudscape) through full SQL support.

Install the information service feature

To change the activity type to information service, we must first install the Information Service optional feature for WebSphere Integration Developer 6.0.2.

If you have not installed the Information Service feature, start the Rational Product Updater (*Help* → *Software updates* → *IBM Rational Product Updater*) and follow the instructions in “Installing the information service feature” on page 582.

Add information service data types


Next, add information service support data types by following these steps:

- ▶ Open the `OrderHandlingFuture1` business process.
- ▶ Right-click on the canvas and select *Create Information Service Support Types*. This adds the information service support types and has to be done the first time you create an information service activity.

The information service support types start with a `t` prefix, for example, `tDataSource` and `tTableName`.

Add a data source variable to the process

We have to create a variable of type `tDataSource` to be used to access the database:

- ▶ Create a variable by selecting the  icon. Name the new variable `ClipsAndTacksDataSource` (Figure 10-78).

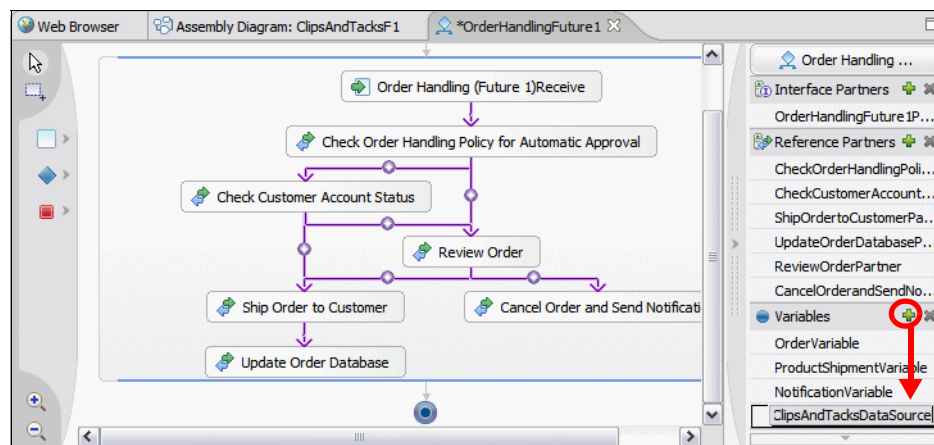


Figure 10-78 Create new variable

- ▶ Select the `ClipsAndTacksDataSource` variable and in the Properties view and select the *Details* tab.

- ▶ Click *Browse* to set the Data Type. Select *tDataSource* (Figure 10-79).

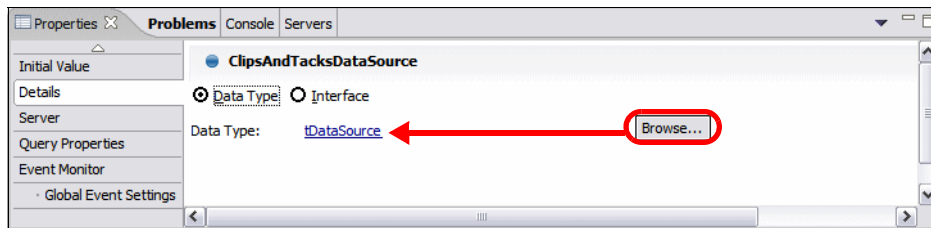


Figure 10-79 Define a *tDataSource* variable

- ▶ Select the *Initial Value* tab and specify the data source JNDI name:
jdbc/cliptack

Change the activity to an information service

Next we change the Update Order Database activity to an information service:

- ▶ Select the Update Order Database activity in the business process editor. Select *Change Type* → *Information Service* (context menu) (Figure 10-80).

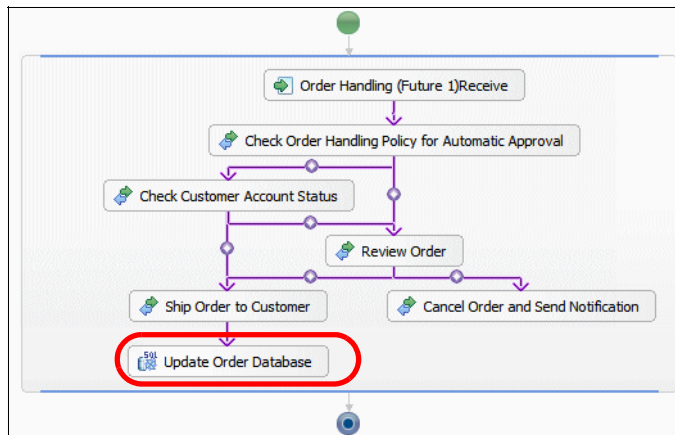


Figure 10-80 Changing an activity to an information service

- ▶ With Update Order Database selected, select the *Details* tab in the Properties view (Figure 10-81):
 - Set the kind of Information Service to *SQL Snippet*.
 - Click *Edit* to set the data source to the `ClipsAndTacksDataSource` variable.
 - Specify the SQL statement as follows:

```
UPDATE CT.ORDERMAIN SET orderstatus = 'SHIPPED' WHERE ordernumber =
```

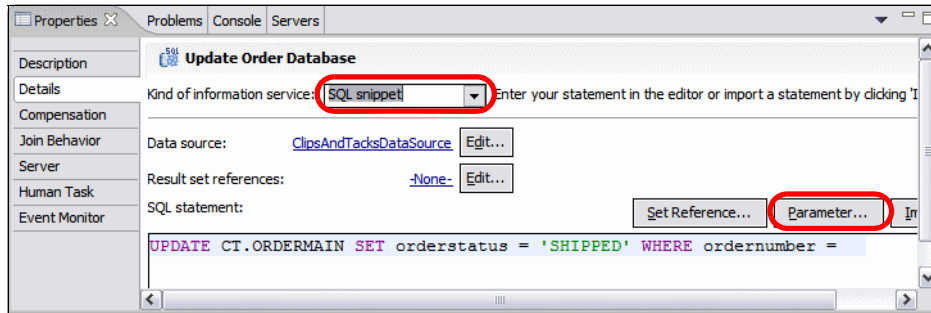


Figure 10-81 Update Order Database: Information service properties

- Click *Parameter* to add a parameter called *orderNumber*, which takes its value from the *OrderNumber* in the *Order* business object (Figure 10-82). Click *OK*.

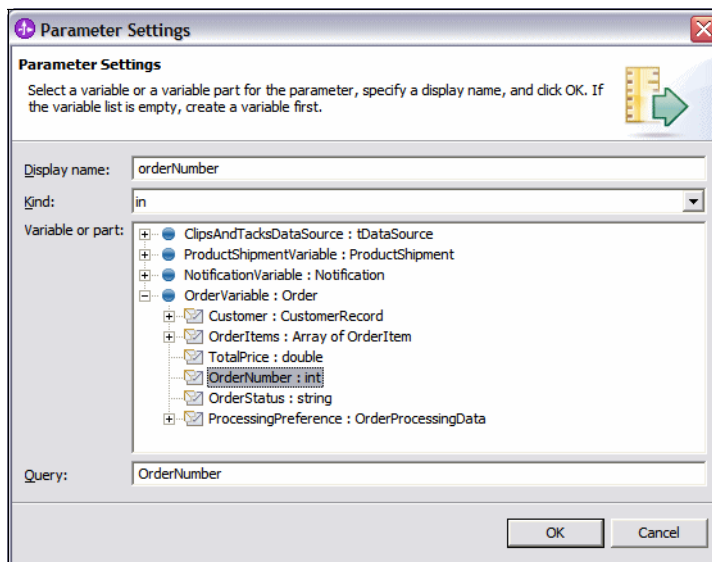


Figure 10-82 Information service parameter settings

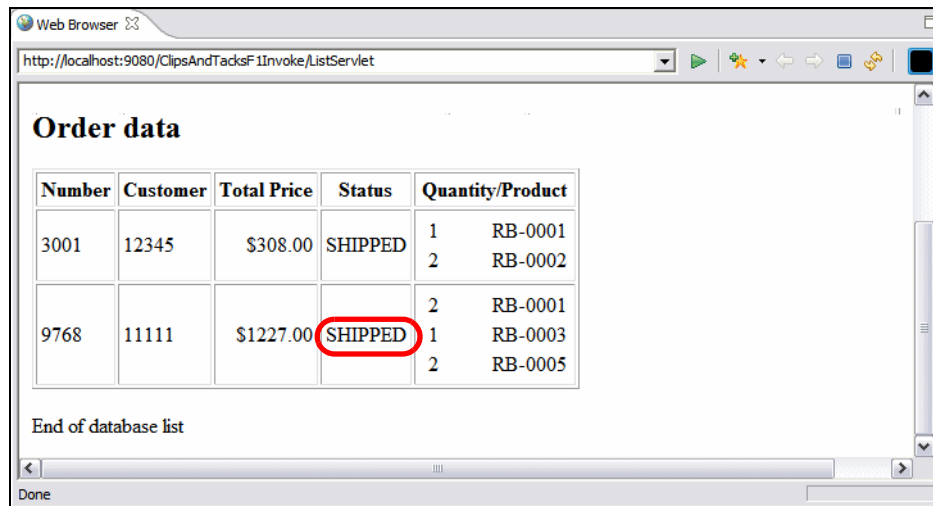
- The complete SQL statement with the parameter inserted is shown here:

```
UPDATE CT.ORDERMAIN SET orderstatus = 'SHIPPED' WHERE
                        ordernumber = #orderNumber#
```
- Save the updates and close the Business Process editor.

Test the information service

To verify that the order status is updated in the order database when an order is shipped, follow these steps:

- ▶ Run the `CreateDatabaseServlet` (*ClipsAndTacksF1Invoke* → *Deployment Descriptor* → *Servlets* → *CreateDatabaseServlet*) and *Run* → *Run on Server*) to initialize the database.
- ▶ Submit an order, then use the human task application to approve and ship the order.
- ▶ Run the `ListServlet` and verify that the status of the order has been updated to *shipped* (Figure 10-83).



The screenshot shows a web browser window with the URL `http://localhost:9080/ClipsAndTacksF1Invoke/ListServlet`. The page displays a table titled "Order data" with the following content:

Number	Customer	Total Price	Status	Quantity/Product
3001	12345	\$308.00	SHIPPED	1 RB-0001
				2 RB-0002
9768	11111	\$1227.00	SHIPPED	2 RB-0001
				1 RB-0003
				2 RB-0005

Below the table, the text "End of database list" is displayed. The status "SHIPPED" for the second order (9768) is circled in red in the original image.

Figure 10-83 Verify database update by Update Order Database activity

Cleaning up the process

The information service is not a supported activity type in the assembly diagram. The activity is still shown as a Java activity.

After converting the database update to use the information service, we can remove the old activity and the old reference from the assembly diagram and from the process:

- ▶ In the Assembly Diagram, delete the `UpdateOrderDatabase` activity.
- ▶ In the Process Diagram, delete the `UpdateOrderDatabasePartner` from the Reference Partners.
- ▶ We suggest leaving the `UpdateOrderDatabase` interface in case we want to change the implementation later.

Using JDBC in a Java activity

Another technique for implementing a database update is using JDBC. We use this method to update the order database when an order is cancelled. Add the following Java code to the Cancel Order and Send Notification activity, immediately above the return statement in the InputCriterion method:

```
SG247148\sampcode\wid\codesnippets\cancelOrderDBUpdate.txt
```

```
try {
    ClipsTacksDAO dao = new ClipsTacksDAO();
    boolean result = dao.updateOrder(Input.getInt("OrderNumber"),
        "DECLINED", false);
    System.out.println("DAO update: " + result);
}
catch (Exception e) {
    System.out.println("DAO update failed: " + e.getMessage());
}
```

Add the following import statement to resolve the references to the ClipsTacksDAO class:

```
import com.clipstacks.dao.ClipsTacksDAO;
```

We must reference the JNDI data source defined in “Creating a data source for the database” on page 218. Follow the following steps to reference the data source from the application’s EJB deployment descriptor:

- ▶ In the Web perspective, open the Deployment Descriptor for the EJB project ClipsAndTacksF1EJB.
- ▶ Select the *References* tab.
- ▶ Select *Module* and click *Add*.
- ▶ Select *Resource reference* as the reference type, then click *Next*.
- ▶ Enter these values (Figure 10-84):
 - Name: ClipsAndTacks
 - Type: javax.sql.DataSource
 - Authentication: Container
 - Sharing scope: Shareable
- ▶ Click *Finish*.

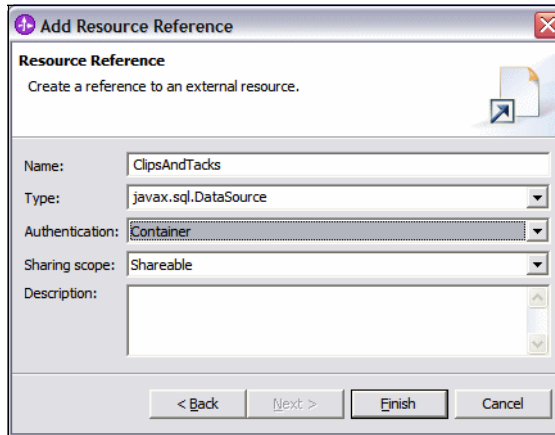


Figure 10-84 Data source resource definition

- In the WebSphere Bindings section, specify the JNDI name for the data source (Figure 10-85):

jdbc/cliptack

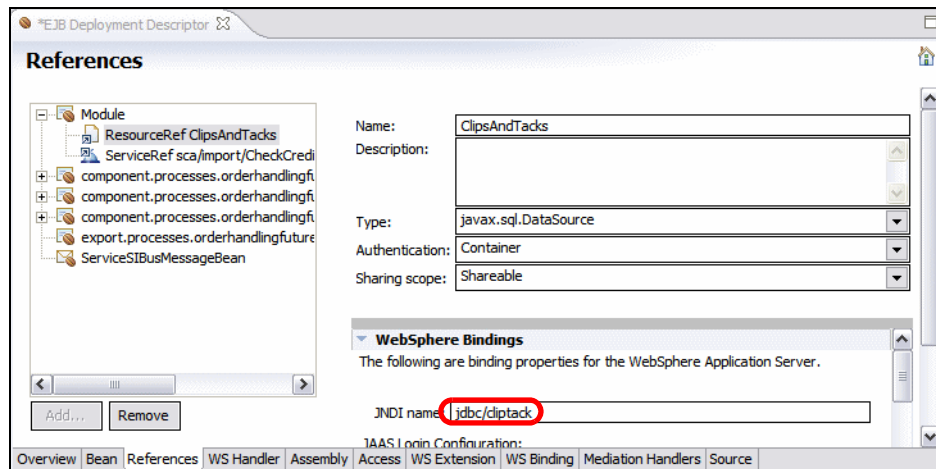


Figure 10-85 EJB Deployment Descriptor: Data source reference

Important: This EJB Deployment Descriptor will be regenerated if you clean your project (*Project* → *Clean*), and this resource definition will be lost. Therefore, if you clean the project, you have to manually re-enter the resource definition for the data source.

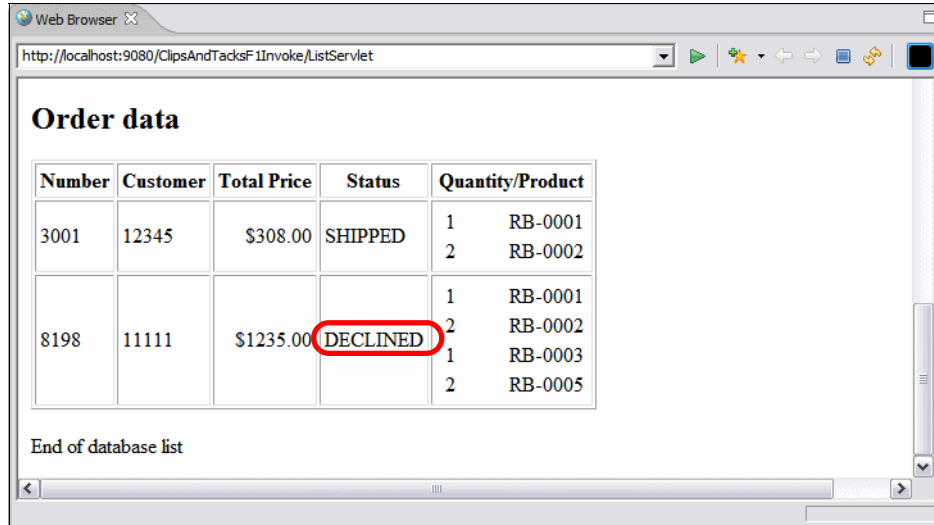
A better approach is to add an information service activity after the Cancel Order and Send Notification activity to update the status of the order. For instructions, see Figure 10-81 on page 290. The SQL statement would be:

```
UPDATE CT.ORDERMAIN SET orderstatus = 'DECLINED' WHERE  
                        ordernumber = #orderNumber#
```

Testing the JDBC update

To verify that the order status is updated in the order database when an order is cancelled, follow these steps:

- ▶ Run the CreateDatabaseServlet to initialize the database (see “Using an information service” on page 287).
- ▶ Submit an order, then use the human task application to *decline* the order.
- ▶ Run the ListServlet and verify that the status of the order has been updated to *declined* (Figure 10-86).



The screenshot shows a web browser window with the URL `http://localhost:9080/ClipsAndTasksF11voke/ListServlet`. The page title is "Order data". Below the title is a table with the following data:

Number	Customer	Total Price	Status	Quantity/Product
3001	12345	\$308.00	SHIPPED	1 RB-0001
				2 RB-0002
8198	11111	\$1235.00	DECLINED	1 RB-0001
				2 RB-0002
				1 RB-0003
				2 RB-0005

Below the table, the text "End of database list" is displayed.

Figure 10-86 Database update by Cancel Order and Send Notification activity

Using the Business Rules Manager

The Business Rules Manager is a Web-based tool that assists the business analyst and other users in modifying business rule values. You can use this Web-based client to browse and edit business rules. The Business Rules Manager is installed in the WebSphere Process Server V6.0.2 test environment installation within Integration Developer.

If you open the administrative console, you can find the BusinessRulesManager application in the list of enterprise applications:

- ▶ Expand *Applications* and select *Enterprise Applications*. The BusinessRulesManager is listed and its status is started.

We can use the Business Rules Manager to test if the template created in “Defining a business rule based on a template” on page 231 works.

- ▶ Submit an order with a value below 750, and it should go directly to shipping.
- ▶ Submit an order with a value between 750 and 800, and it should go to the Order Review activity.
- ▶ Use the human task application to process the orders.

Using the Business Rules Manager

Select the server in the Servers view and *Launch* → *Business Rules Manager*.

- ▶ The Business Rules Manager opens. Expand the sections.
- ▶ Select *AutomaticApprovalF1* under Rule Books (Figure 10-87).

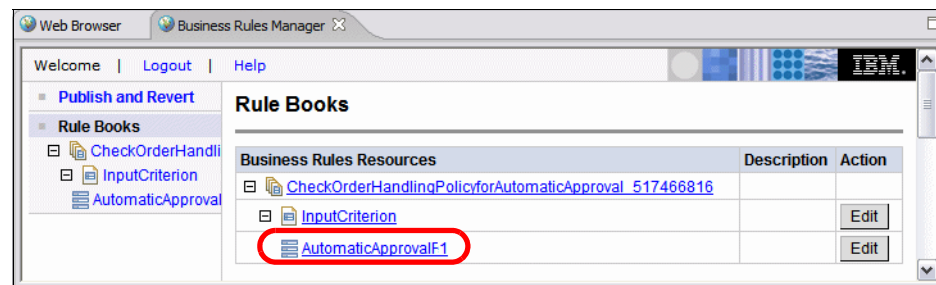


Figure 10-87 Business Rules Manager: Rule books

- ▶ Click *Edit* (Figure 10-88).

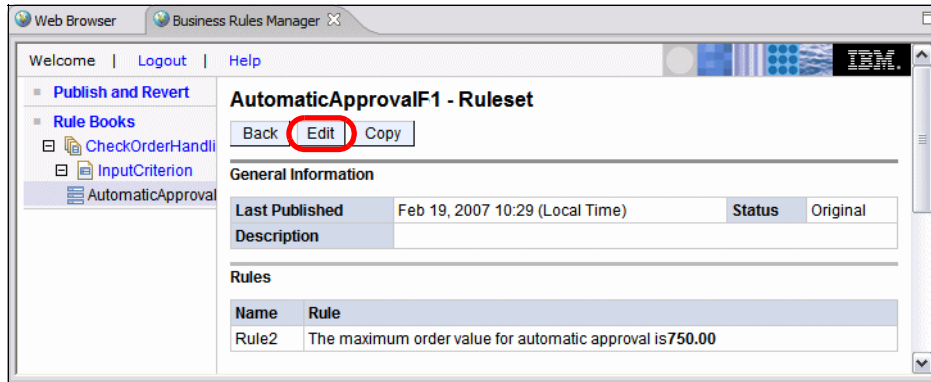


Figure 10-88 Business Rules Manager: Ruleset

- ▶ Change the value to 800.00. It is flagged red—not allowed—because we set the range to be between 700.00 and 800.
- ▶ Change the value to **799.00**, then click *Save*. Notice the message: "AutomaticApprovalF1" has been temporarily saved. You may publish the changes from the "Publish and Revert" page.
- ▶ Select *Publish and Revert* on the left hand side and click *Publish* (Figure 10-89).

The message Selected rule page(s) have been published successfully is displayed.

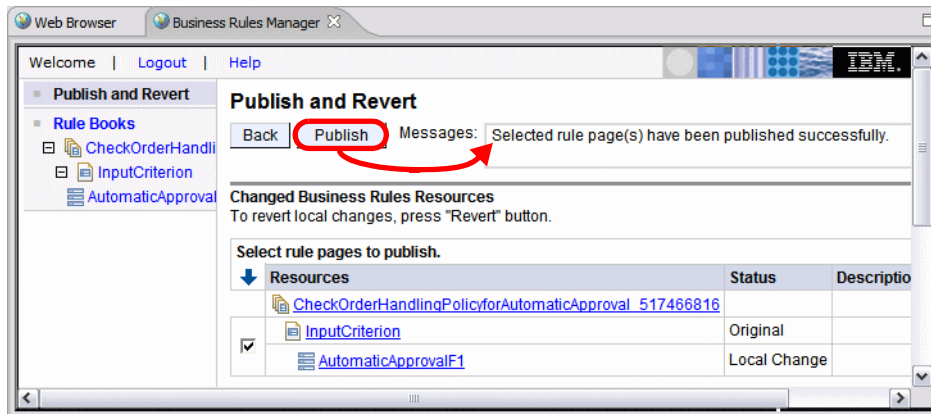


Figure 10-89 Publishing a business rule

The updated business rule is now active. Submit an order with a total price between \$750 and \$799. This order goes directly to shipping! This proves that the updated business rule is active.

Exporting business rules

Business rules can be copied from one server to another by export import in the administrative console:

- ▶ Open the administrative console.
- ▶ Expand *Servers* → *Application servers*, and select *server1*.
- ▶ Select *Business Integration* → *Business Rules* → *Business rules*.
- ▶ Select one or more rules and click *Export* (Figure 10-90).

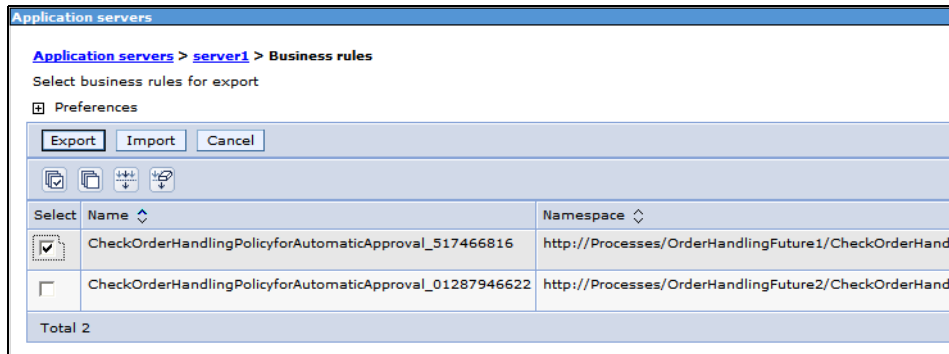


Figure 10-90 Exporting business rules: Selection

- ▶ A zip file is generated for download (Figure 10-91).

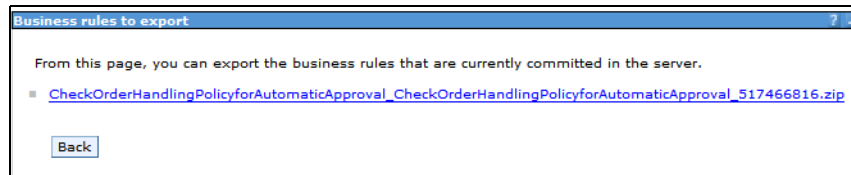


Figure 10-91 Exporting business rules: Resulting zip file

- ▶ The zip file contains the rule definitions and the runtime data (Figure 10-92).

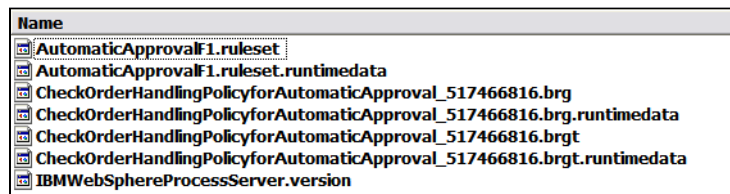



Figure 10-92 Exporting business rules: Resulting content

- ▶ The zip file can be imported into another system.

Using the Debugger

The Debugger built into the Eclipse platform and Integration Developer can debug Java code as well as business processes.

To make use of the Debugger during execution of a business process, the WebSphere Process Server must be started in Debug mode. Select the server and *Debug* or click the *Debug* icon .

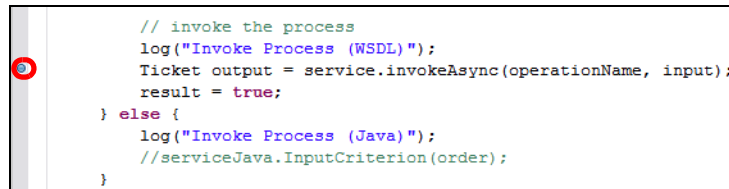
Wait for the server to be ready. If the projects are not deployed, add them to the server (ClipsAndTacksFlApp, ClipsAndTacksFxHumanCustomEAR, and CreditServiceEAR).

Setting breakpoints

To stop execution of the application at certain points, insert breakpoints into the application.

Breakpoint in Java

Open a Java class, for example the invocation class `InvokeOrderHandling`, and set a breakpoint just before invocation of the business process (Figure 10-93).



```
// invoke the process
log("Invoke Process (WSDL)");
Ticket output = service.invokeAsync(operationName, input);
result = true;
} else {
log("Invoke Process (Java)");
//serviceJava.InputCriterion(order);
}
```

Figure 10-93 Setting a breakpoint in Java

To set the breakpoint, double-click into the border or right-click into the border and select *Toggle Breakpoint*.

Breakpoint in the business process

Open the process diagram. Select a task and *Debug* → *Add Entry Breakpoint* (or *Add Exit Breakpoint*). You can add many breakpoints into the business process to control the flow and monitor variable values (Figure 10-94).

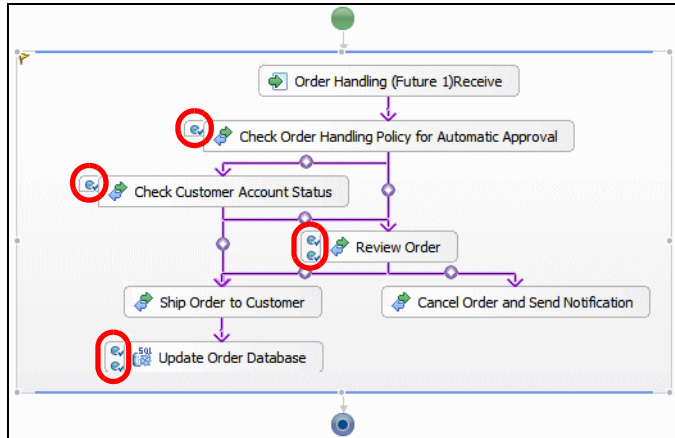


Figure 10-94 Setting breakpoints in the business process

Debugging the application

Run the application in the usual way by submitting orders using the Web front-end. Once you submit an order, the breakpoint in the Java class stops the execution. You are prompted to switch to the Debug perspective.

The Debug perspective consists of a number of panes (Figure 10-95). You can see in which class the application stopped in the Debug view, the variable values in the Variables view, and the Java code with the breakpoint. The Breakpoints view shows the list of all the breakpoints.

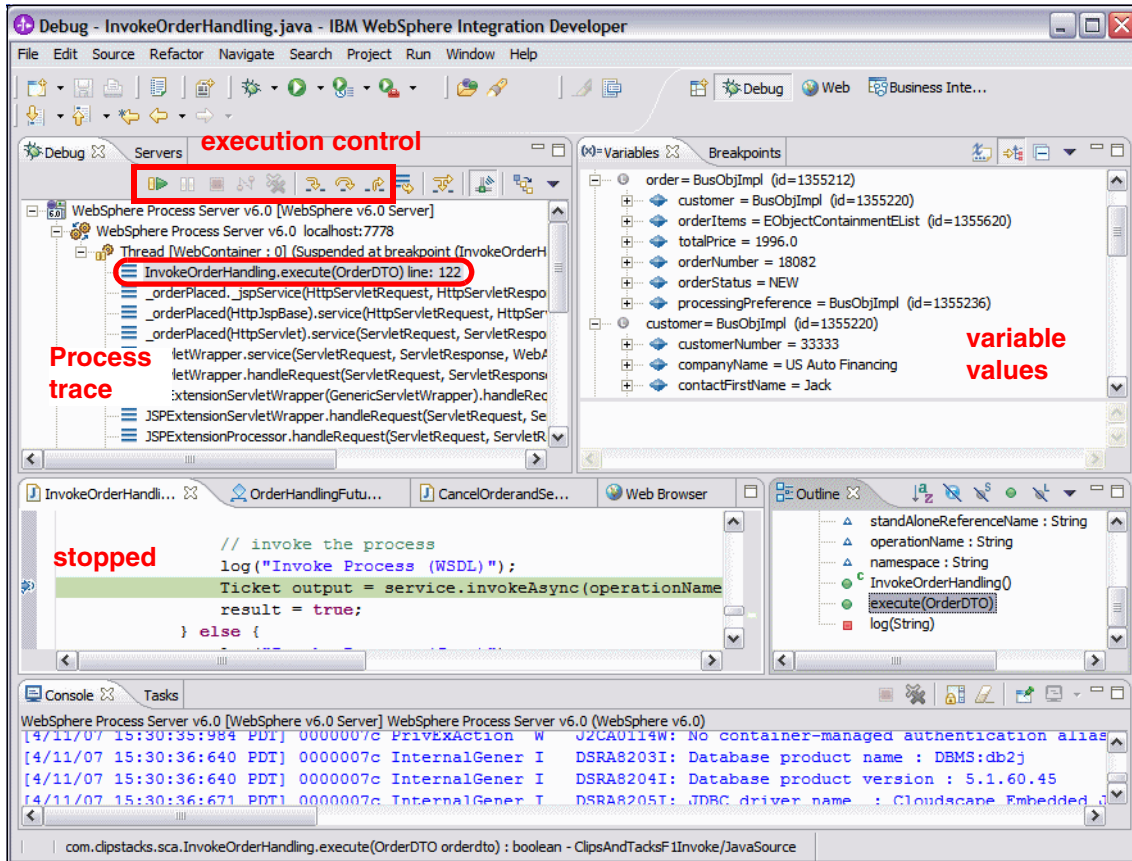






Figure 10-95 Debug perspective: Application stopped at Java breakpoint

Using the execution control icons, you can continue the execution to the next breakpoint , step into a line of code , step over a line of code , or step to the return of the current method .

Changing a variable value

You can change the values of the variables by selecting a variable and *Change Value*. You are prompted to enter a new value for the variable. This enables you to control the flow of the execution, where it is dependent on the value of certain variables.

Stopping at a breakpoint in the business process

When a breakpoint in the business process is reached, the Debug perspective displays the process diagram with a yellow marker where the execution is stopped (Figure 10-96).

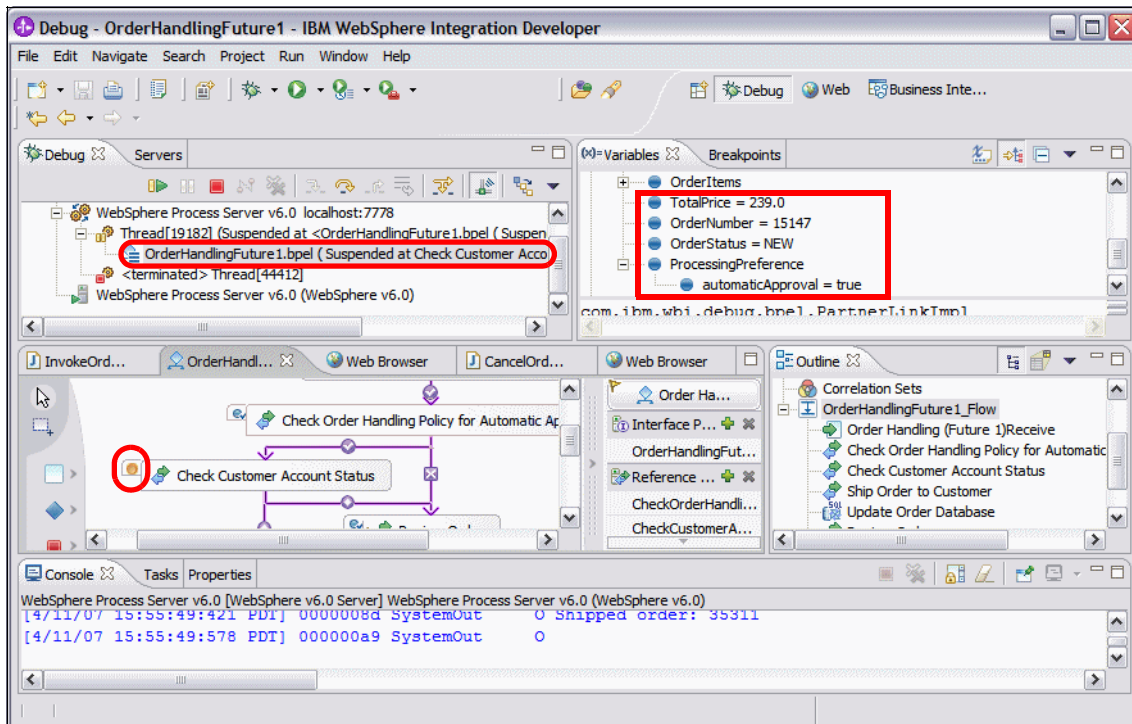


Figure 10-96 Debug perspective: Application stopped at process breakpoint

Handling of human tasks

Do not forget to handle the human tasks using the custom human task application or the BPC Explorer. When the process is waiting for a human task, a small hourglass is displayed in the process diagram (Figure 10-97).

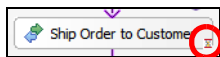


Figure 10-97 Debug perspective: Waiting for a human task

Use the Debug perspective and breakpoints when the application does not run through the expected path. By stopping at suitable points and inspecting the variables, you can find the cause for the unexpected behavior.

Summary

In this chapter we described in detail how to implement the business process application after importing the BPEL and associated files generated by WebSphere Business Modeler.

We showed how to implement a business rule group, Java activities, and human tasks. Then we implemented a Web front-end to invoke the business process. We showed how to manage human tasks using the BPC Explorer without or with customized JSPs.

Then we showed how to invoke an external Web service from an activity, and how to implement a stand-alone application for human task management. We showed two different techniques for updating the order database.

Finally we showed how to use the Business Rule Manager to update a business rule at runtime, and how to use the Debugger to debug the business process and change variable values.

Important: In a real development environment we would use team development with a repository, such as Rational ClearCase® or Concurrent Versions System (CVS).



Implementing security

In this chapter we implement security in the server and in our applications.

First we implement a simple security setup to control which users can invoke the human tasks to handle the orders.

Then we use an LDAP server to implement the same security specifications for a production environment.

Implementing security for human tasks

In this section we describe how to set up a simple security environment to control who can use the human task functions of reviewing orders, shipping orders, and canceling orders.

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 the user registries that 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 into your system.
- ▶ **Lightweight Directory Access Protocol (LDAP)**—When working with multiple servers, such as clustering and high availability, and with multiple systems, you will probably run into limitations using a single system'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 for user registries is a user registry that is not one of the operating system user registries supported by WebSphere and is not an LDAP-compliant user registry. Connecting to these types of user registries is an advanced but supported option, and requires a software development effort to complete.

Using a custom user registry for security

We explore the custom user registry option to implement security for the human task application. In “Using an LDAP server” on page 318, we will configure an LDAP server to implement the same security constraints.

Defining groups and users

To implement security, we need groups and users. For our application, we require an order manager, a shipper, and an administrator. Therefore, we define the groups and users shown in Table 11-1. The user ID **wid** will be used to start and stop the server.

Table 11-1 Groups and users for human task security

Group	Users
administrator	wid, ueli, db2admin ^a
ordermanager	andre, ueli
shipper	russ, larissa, ueli

a. We added db2admin to run the Monitor test environment with security.

A custom registry requires a Java program that implements `UserRegistry`, which defines methods such as `initialize`, `checkPassword`, `getUsers`, `getGroups`, `getUsersForGroup`, `getUniqueGroupId`, and so forth.

Implementation

Our implementation uses two property files and a Java program provided in:

SG247148\sampcode\security

- **groups.props**—Property file with group definitions:

```
# Format:
# name:gid:users:display name
# where name = groupId of the group
#       gid  = uniqueId of the group
#       users = list of all the userIds that the group contains
#       display name = (optional) display name for the group
administrator:001:wid,ueli,db2admin:Administrative group
ordermanager:002:andre,ueli:Order Manager group
shipper:003:russ,larissa,ueli:Shipper group
```

- **users.props**—Property file with user definitions:

```
# Format:
# name:passwd:uid:gids:display name
# where name = userId/userName of the user
#       passwd = password of the user
#       uid    = uniqueId of the user
#       gid    = groupIds of the groups that the user belongs to
#       display name = (optional) display name for the user
wid:wid:101:001:WID Admin
ueli:ueli:102:001,002,003:Ueli Wahli
andre:andre:103:002:Andre Venancio
russ:russ:104:003:Russell Scher
larissa:larissa:105:003:Larissa Leybovich
db2admin:xxxxxxx:106:001:DB2 Administrator
```

- **FileRegistrySample**—Java program that contains the code to read the property files and implement all the required methods.

Placement of the user registry files

We put the property files into the directory pointed to by the WebSphere Application server variable `${USER_INSTALL_ROOT}`. In Integration Developer this points to `<WID-HOME>/pf/wps`:

```
<WID-HOME>/pf/wps/UserCustomRegistry/groups.props
```

The `FileRegistrySample` program and subclass `RegExpSample` must be in the class path of the server. We put the program into:

```
<WID-HOME>/runtimes/bi_v6/lib/ext
```

Specifying security in the Process Server

Enabling security in the Process Server entails a sequence of steps:

- ▶ Configuring security for submitting events
- ▶ Defining security using a user registry
- ▶ Enabling global security
- ▶ Specifying security in Integration Developer
- ▶ Restarting the server

Configuring security roles for submitting events

The order process is started using a customer GUI, and the user credentials used for accessing the GUI are used to authenticate event submission. Because this user interface is a Web-based application that any Internet user can access, we have to configure the event server to accept that the creator of the event is any unauthenticated user.

To allow any user to submit events, perform the following steps:

- ▶ Open the administrative console (select the server and *Run administrative console*) and navigate to *Applications* → *Enterprise Applications* → *EventServer*.
- ▶ From the Additional Properties, select *Map security roles to users/groups* (Figure 11-1).

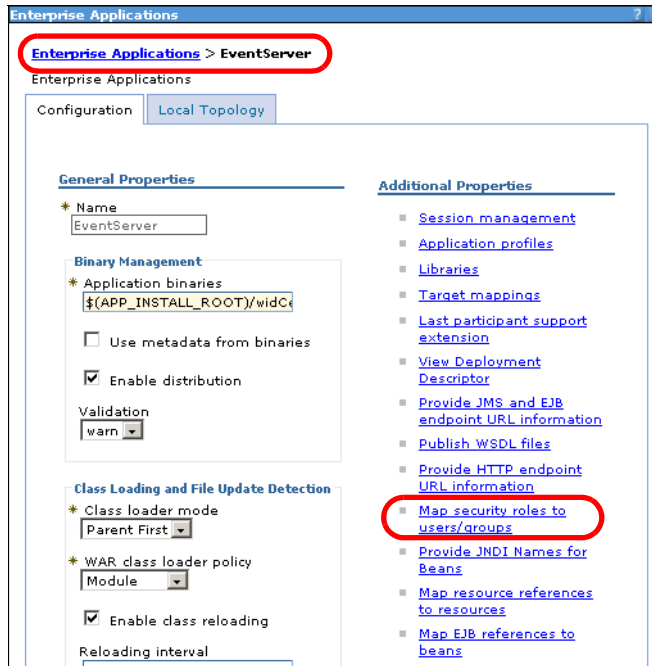


Figure 11-1 Mapping security roles for the event server

- ▶ Select *Everyone* for the eventCreator entry (Figure 11-2).

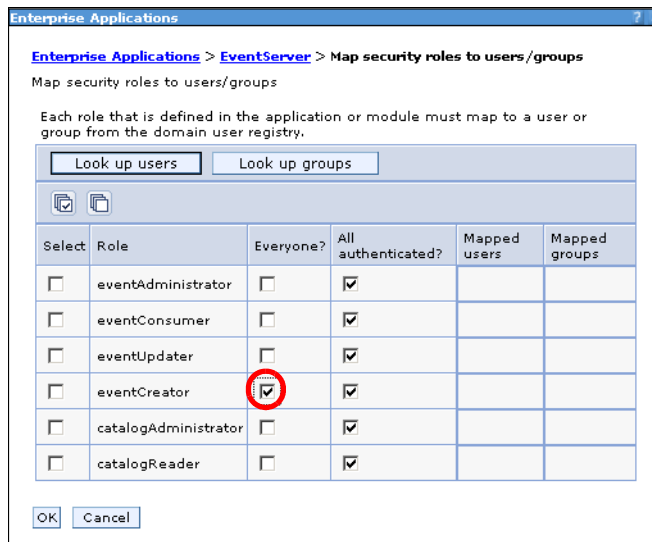


Figure 11-2 Specifying that everyone can create events

Define security using a user registry

In the administrative console, expand *Security* → *Global Security* (Figure 11-3).

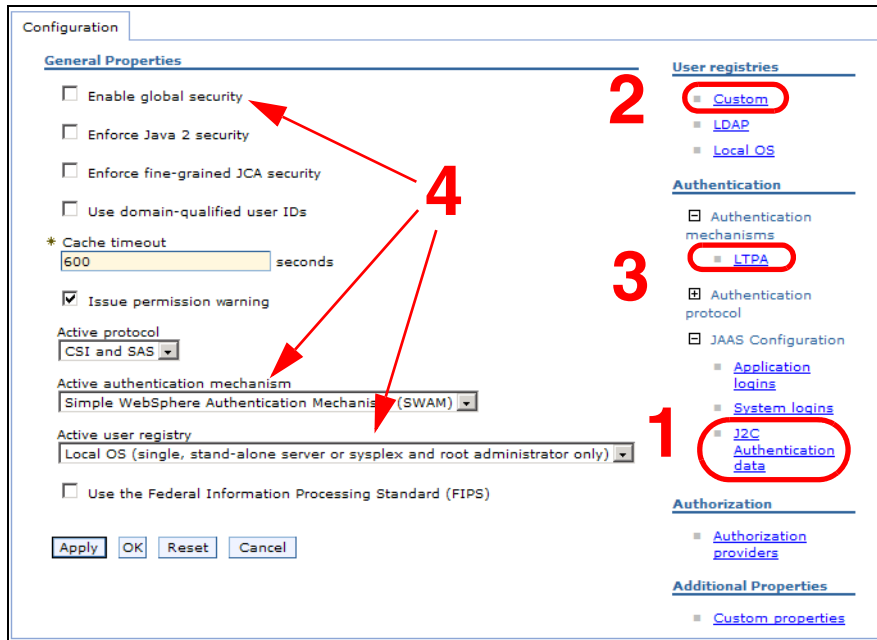


Figure 11-3 Global security: initial

1. 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` and the encrypted password `wid` (Figure 11-4).

Select	Alias	User ID	Description
<input type="checkbox"/>	BPCEventCollectorJMSAuthenticationAlias_widNode_server1	wid	Authentication alias for Business Process Choreographer Event Collector
<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

Figure 11-4 J2C authentication data entries (abbreviated)

Note that we defined the user ID `wid` in `users.props` with the password `wid`. If you use a different password, you must update the aliases with your password.

2. Back in Global Security, select *Custom* under *User registries* (right side):
 - Enter `wid` as user ID and password.
 - The custom registry class name points by default to our program name `com.ibm.websphere.security.FileRegistrySample`.
 - Select *Ignore case for authorization* (optional).
 - Click *Apply* (Figure 11-5).

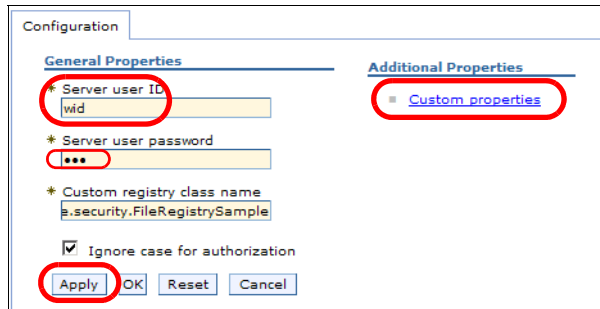


Figure 11-5 Custom registry definition

- Click *Custom properties*. Add two custom properties named `groupsFile` and `usersFile`, matching the code in the `FileRegistrySample` program. The two properties point to our files (Figure 11-6):

```
${USER_INSTALL_ROOT}/UserCustomRegistry/groups.props  
${USER_INSTALL_ROOT}/UserCustomRegistry/users.props
```

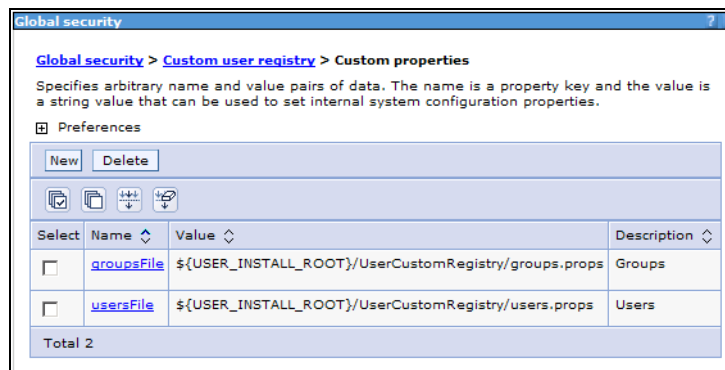


Figure 11-6 Custom registry properties

- Back in Global Security, select *Authentication mechanism* → *LTPA* under *Authentication* (right side). Enter the password wid twice and click *Apply* (Figure 11-7).

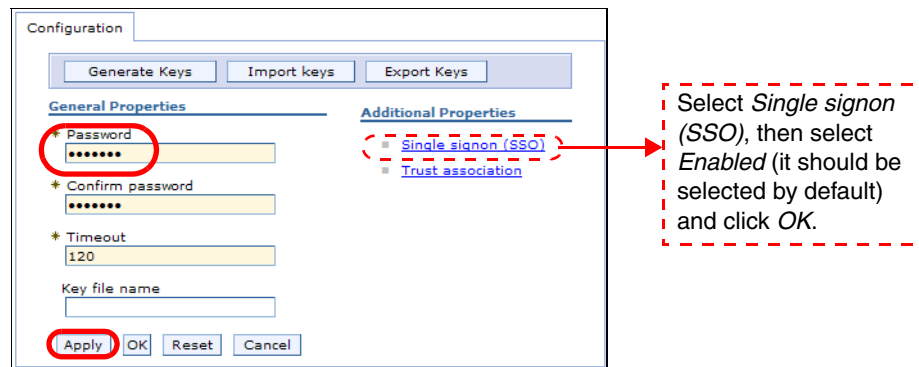


Figure 11-7 Authentication using LTPA

Enable global security

- Back in Global Security (Figure 11-8).
 - Select *Enable global security*, and deselect *Enforce Java 2 security*.
 - For active authentication mechanism, select *Lightweight Third Party Authentication (LTPA)*.
 - For active user registry, select *Custom user registry*.

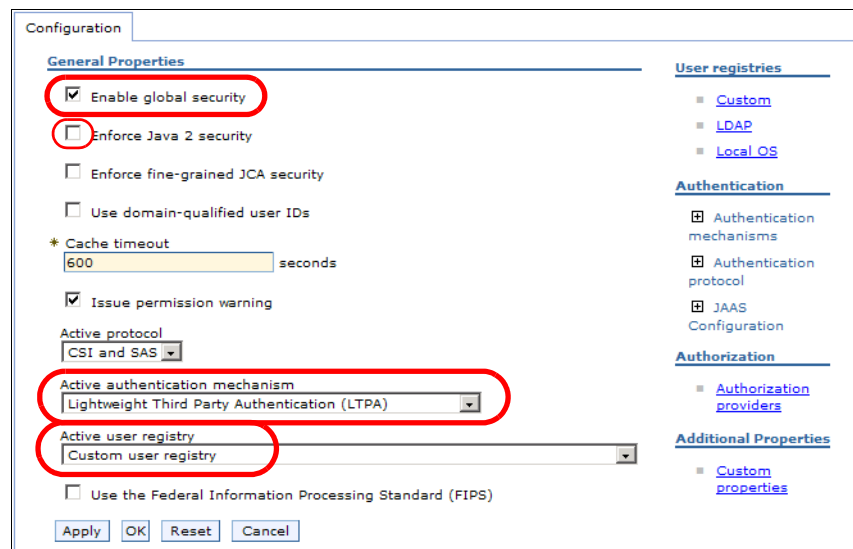


Figure 11-8 Enable global security

- Click *OK*. The server verifies the custom registry files.

Save the configuration

Click *Save* to save the configuration changes. then click *Logout* in the administrative console.

Specify security in Integration Developer

Because the server is started and stopped from Integration Developer, we have to specify user ID and password in the server configuration:

- ▶ Select the server in the Servers view and *Open* (Figure 11-9).

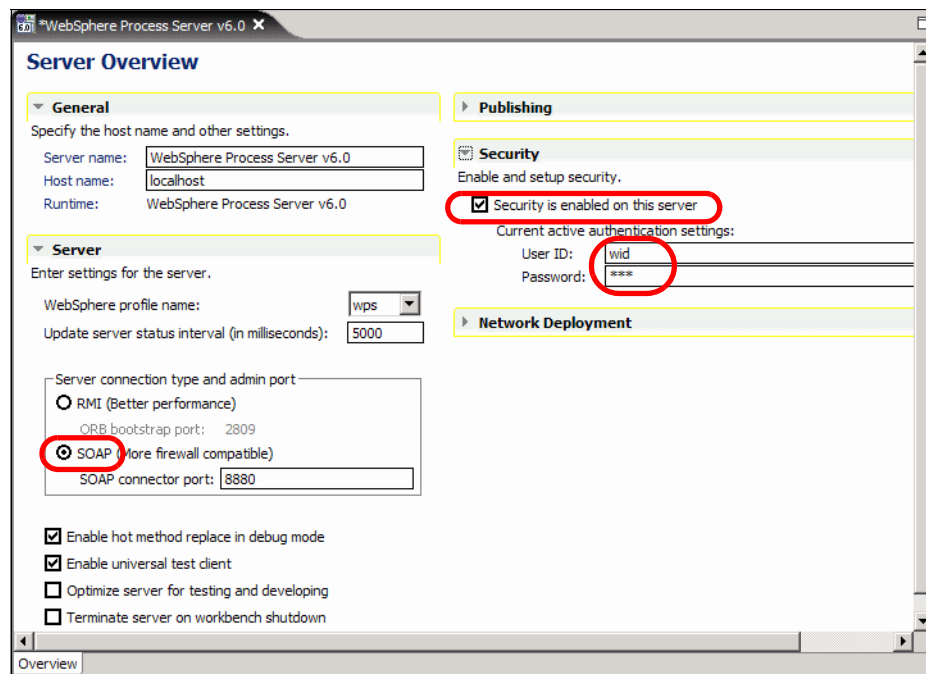


Figure 11-9 Server configuration in Integration Developer

- Select *SOAP* as the connection type, and enter port 8880.
- Expand *Security* and select *Security is enabled on this server*. Enter *wid* as user ID and password. This must match what you specified in the server custom registry.
- Save and close the configuration.


Restart the server

Stop and start the server. You should see the console when the server is ready.

Start the administrative console. You have to accept a certificate and then you have to login with the specified user ID (wid) and password.

Using the groups for human tasks

To enable security for the human tasks we have to configure each human task with the correct group:

- ▶ In the Business Integration view expand *ClipsAndTasks* → *Business Logic* → *Human Tasks*.
- ▶ Open the *ReviewOrder* task (Figure 11-10):
 - For receiver settings, click the *Potential Instance Creator* icon  to add an entry to the list.
 - Select *Potential Instance Creator* and verify in the Properties view that the verb is set to *Everybody*. (Instances of the process are created by unauthenticated Web users submitting orders.)

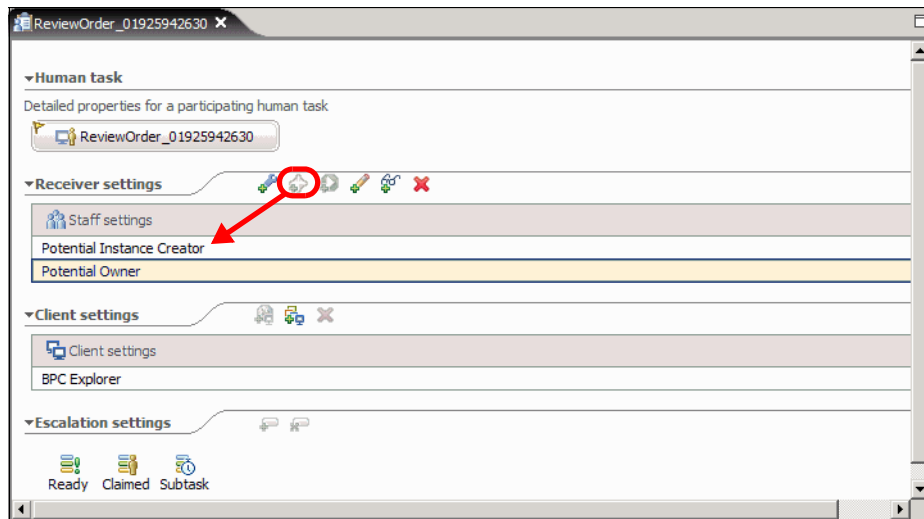


Figure 11-10 Human task configuration

- Select *Potential Owner* and in the Properties view, select *Group Members* as the verb (Figure 11-11).

For parameters, enter **ordermanager** as group name, select *false* for subgroups, and enter **administrator** for alternative group name 1.

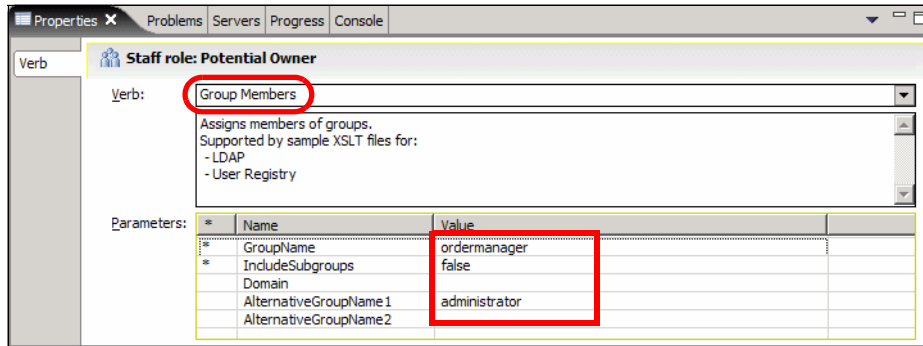


Figure 11-11 Human task properties

Only an order manager or the administrator will be able to claim (work on) a review order task.

- ▶ Open the ShipOrderToCustomer task and perform the same configuration changes:
 - Add the Potential Instance Creator with Everybody as verb, and set the Potential Owner to *Group Members*.
 - However, enter **shipper** as group name, select *false*, and enter administrator as alternative group name.

Only a user from the shipper group, or an administrator, can claim a shipping task.

Verify the staff plug-in

With the human task open in the editor, select the task itself and in the Properties view, Details tab, verify that the JNDI name points to the user registry configuration (Figure 11-12).

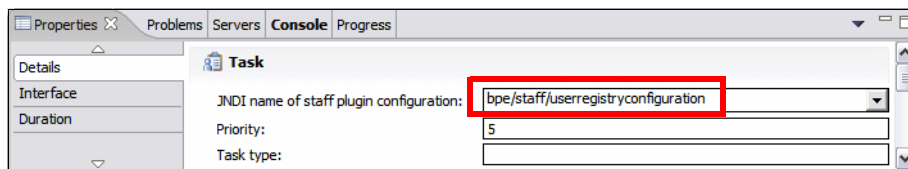


Figure 11-12 JNDI name for the staff plug-in

Restart the application

Select the server in the Servers view and *Restart Project* → *ClipsAndTacksF1App*.

Testing human task security with the BPC Explorer

Redeploy the application to the server. Then submit orders using the Web front-end.

Once you have orders waiting for human interaction, start the BPC Explorer by selecting the server and *Launch* → *BPC Explorer*:

- ▶ Accept the security certificate.
- ▶ You are prompted to login (Figure 11-13).

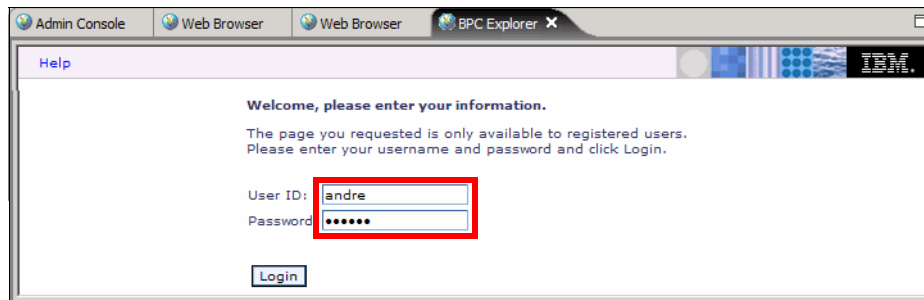


Figure 11-13 BPC Explorer: login

- ▶ Login as order manager (andre). You will only see review (ReviewOrder) tasks. You can claim and complete the tasks by clicking *Work on*.
- ▶ Click *Logout*, then login as a shipper (russ). You will only see shipping (ShipOrdertoCustomer) tasks, which you can claim and complete.
- ▶ Click *Logout*, then login as an administrator (ueli). You can see all human tasks waiting for interaction.

Using the human task Web application with security

The customized human task application does show the human tasks after configuring security. However, when you try to claim a task for processing, you get this error:

```
Error: Exception CWTKA0068E: User 'UNAUTHENTICATED' is not allowed to perform the requested action 'claim()' on task .....
```

We have to implement security in the Web application. See “Implementing a customized human task application” on page 279 for the description of the application.

Implementing security in the human task Web application

First we have to provide a login page so that users can authenticate using the user IDs provided:

- ▶ **login.jsp**—Prompts the user for user ID and password. This is a standard facility of J2EE and uses the `j_security_check` servlet provided by the application server.
- ▶ **loginError.html**—This error page is displayed when the login fails.

Note: The imported Web application is preconfigured for security. Follow these steps to verify the setup.

Next we have to configure security in the Web application deployment descriptor:

- ▶ Open the deployment descriptor of the `ClipsAndTacksFxHumanCustomWeb` project and select the *Security* tab. Figure 11-14 shows the completed definitions.

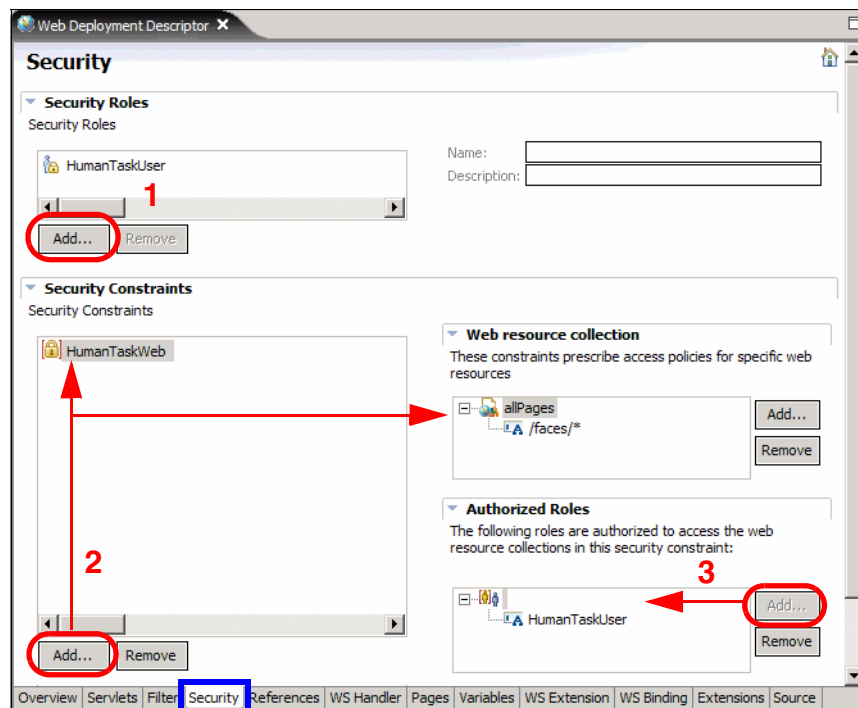


Figure 11-14 Web deployment descriptor with security

1. Click *Add* to define the HumanTaskUser security role.
 2. Click *Add* to define the HumanTaskWeb security constraint. In the dialog enter allPages as resource name and /faces/* as ULR pattern. This protects all the pages that deal with the human tasks.
 3. Select the allPages resource, click *Add* for authorized roles, and select the HumanTaskUser role.
- ▶ Select the *Pages* tab. For login authentication method, select *FORM*. Then select the login.jsp and the loginError.html page for the login and error pages (Figure 11-15).
- You can also delete all the welcome pages except for index.html.

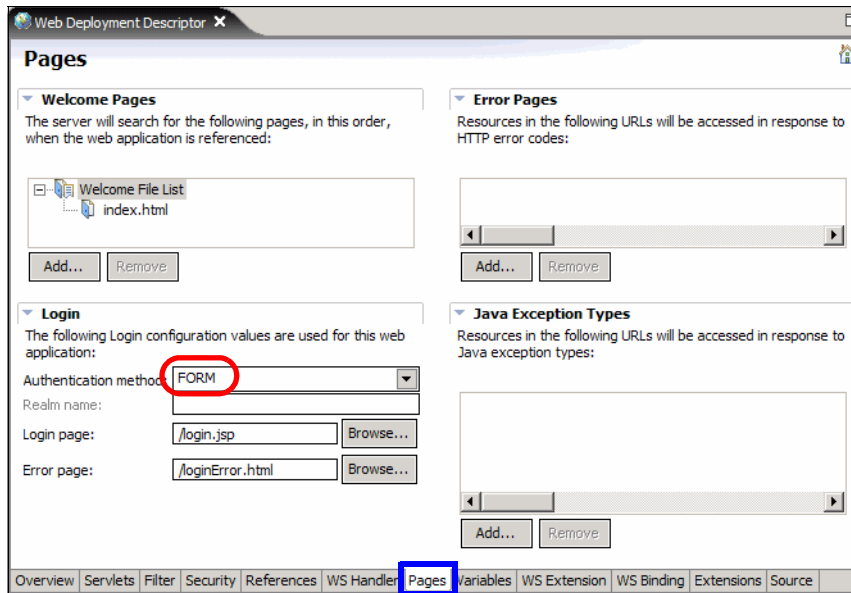


Figure 11-15 Web deployment descriptor login pages

- ▶ Save and close the deployment descriptor.

Now we have to update the enterprise application deployment descriptor:

- ▶ Open the deployment descriptor of the ClipsAndTasksHumanCustomEAR application and select the *Security* tab.
- ▶ Click *Gather*, then select the HumanTaskUser and select *All authenticated users* (Figure 11-16). Save and close the deployment descriptor.

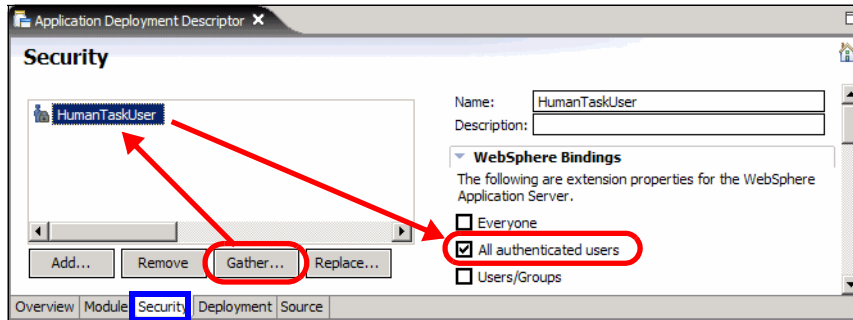


Figure 11-16 Enterprise application deployment descriptor security

Running the human task application with security

After submitting some orders, open the human task application:

`https://localhost:9443/ClipsAndTacksF1HumanCustomWeb/`

- ▶ Accept the security certificate.
- ▶ You are prompted to login. Login as order manager andre/andre1 (Figure 11-17).

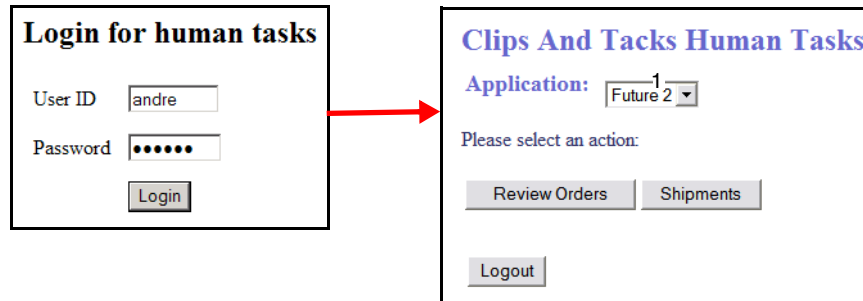


Figure 11-17 Human task application login

- ▶ Click *Review Orders* and you can process orders waiting for review. If you click *Shipments* you always get the message that no tasks are waiting.
- ▶ Click *Logout* and login as a shipper (russ/russ1) and you can process shipments.
- ▶ Login as administrator (ue1i/ue1i1) and you can process all tasks.

We have now implemented security for the human tasks and only authorized users can process the orders waiting for interaction.

Using security in the generated user interface application

The generated user interface application is preconfigured for security with a security role (`CustomClientUser`), login page (`Login.jsp`), and error page (`LoginError.jsp`).

The only action to be performed is updating the enterprise application deployment descriptor by gathering the security specification. Refer to Figure 11-16 on page 317 for instructions.

Using an LDAP server

Instead of using a custom user registry, we can also use an LDAP server. We installed IBM Tivoli Directory Server V5.2 as our LDAP server. Refer to “Installation of IBM Tivoli Directory Server V5.2” on page 612.

Groups and users

We configured the server with the groups and users for our scenario. Refer to “Directory server configuration” on page 614 and “Groups and users” on page 617.

Notice the structure of the groups and users in our LDAP server:

- ▶ There are two top level containers: `cn=groups` and `cn=users`.
- ▶ The groups (`cn=ordermanager`, `cn=shipper`, `cn=administrator`) are under the `cn=groups` container.
- ▶ The users (`uid=wid`, `uid=andre`, and so forth) are under the `cn=users` container.

This structure is important for setting up the LDAP queries.

Configuring the Process Server for LDAP

To run the server with LDAP, we have to configure global security and the human task plug-in for LDAP. Open the administrative console and perform the following steps.

Staff plug-in provider

The staff plug-in provider for LDAP must be configured for the LDAP server:

- ▶ Select *Resources* → *Staff plug-in provider*.
- ▶ Click *LDAP Staff Plugin Provider* (Figure 11-18).

Select	Name	Description
<input type="checkbox"/>	LDAP Staff Plugin Provider	This staff plugin provider can be used for LDAP based staff queries.
<input type="checkbox"/>	System Staff Plugin Provider	This staff plugin provider may be used for System based staff queries.
<input type="checkbox"/>	User Registry Staff Plugin Provider	This staff plugin provider can be used for User Registry based staff queries.
Total 3		

Figure 11-18 LDAP staff plugin provider

- Under Additional Properties, click *Staff plug-in configuration*. Notice the JNDI name and the XSL Transform File (Figure 11-19):

```
bpe/staff/sampleldapconfiguration
${WBI_INSTALL_ROOT}/ProcessChoreographer/Staff/LDAPTransformation.xsl
```

Select	Name	Description	JNDI name	XSL Transform File
<input type="checkbox"/>	LDAP Staff Plugin Configuration sample	This sample staff configuration can be used for LDAP based staff queries.	bpe/staff/sampleldapconfiguration	\${WBI_INSTALL_ROOT}/ProcessChoreographer/Staff/
Total 1				

Figure 11-19 LDAP staff plug-in provider details

- Click *LDAP Staff Plugin Configuration sample*, then click *Custom properties* under Additional Properties.

- The BasedN property should be set to `dc=ibm,dc=com`.
- Select the property `ProviderURL` and change the value of the host name from `localhost` to your LDAP server (Figure 11-20):

```
From: ldap://localhost:389
To: ldap://KLCHL2Y:389          host name of the LDAP server
```

- Click *OK*.

BaseDN	dc=ibm,dc=com	Default base dn for LDAP search operations. Sample: 'o=acme,c=us'.
CasesensitivityForObjectclasses	true	When set to true, LDAP objectclass names will be treated in a case-sensitive manner (e.g. inetOrgPerson and inetorperson will be treated as different names).
ContextFactory	com.sun.ldap.LdapCtxFactory	The name of the JNDI contextFactory Java class. Sample: 'com.sun.ldap.LdapCtxFactory'.
ProviderURL	ldap://KLCHL2Y:389	The provider URL of the LDAP/JNDI server to connect to. Sample: 'ldap://localhost:389'.

Figure 11-20 LDAP provider URL

Enable global security for LDAP

We change the global security settings to use LDAP instead of the custom registry:

- ▶ Select *Security* → *Global security*.
- ▶ Under User registries, select *LDAP* (top-right).
- ▶ Enter the values as shown in Figure 11-21.

Configuration

General Properties

- * Server user ID: wid
- * Server user password: *****
- Type: Custom
- * Host: KLCHL2Y
- Port: 389
- Base distinguished name (DN): dc=ibm,dc=com

Bind distinguished name (DN):

Bind password:

Search timeout: 120 seconds

Reuse connection

Ignore case for authorization

SSL enabled

SSL configuration: widNode/DefaultSSLSettings

Apply OK Reset Cancel

Additional Properties

- [Advanced Lightweight Directory Access Protocol \(LDAP\) user registry settings](#)
- [Custom properties](#)

Figure 11-21 LDAP security configuration

- Enter the LDAP server user ID and password.
 - Select the LDAP type (we selected *IBM Tivoli Directory Server*).
 - Enter the host name where the LDAP server runs.
 - Enter the base distinguished name.
 - Click *Apply*.
- ▶ Still in the LDAP User Registry panel under Additional Properties, click *Advanced Lightweight Directory Access Protocol (LDAP) user registry settings*.
- ▶ We want to be able to look up users by their user ID, so change the setting for User filter (Figure 11-22):

From: (&(uid=%v)(objectclass=ePerson))
 To: (&(uid=%v)(objectclass=inetOrgPerson))

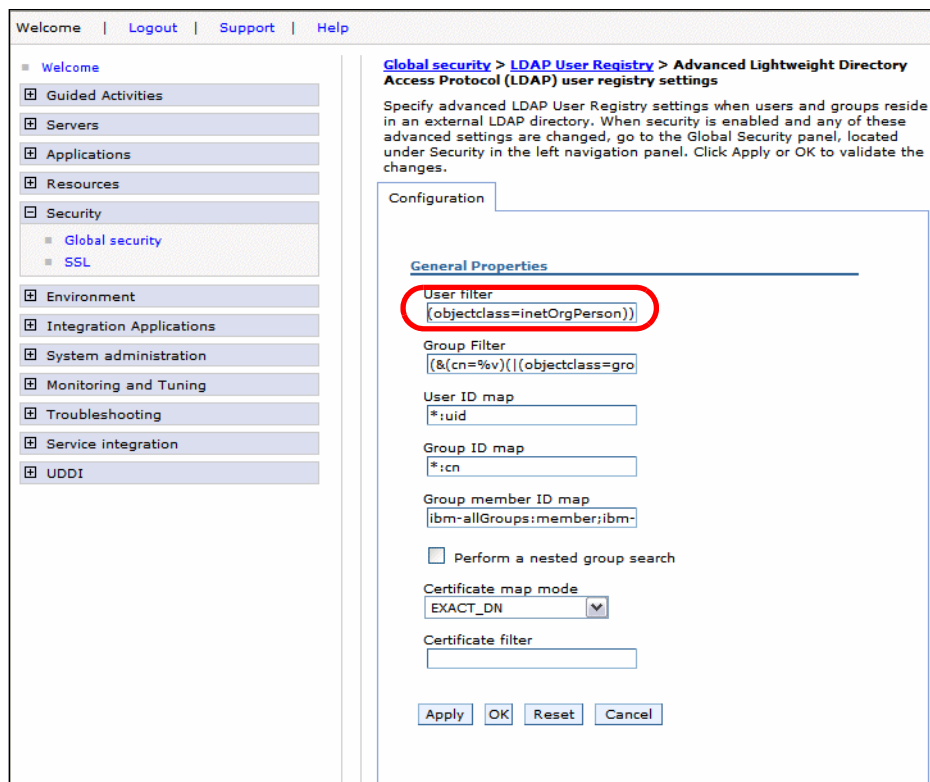


Figure 11-22 Enable security: Advanced LDAP user registry settings

- ▶ Click *OK* twice to return to the main Global security panel.

Note: Once we change the User filter, the LDAP type will change from IBM Tivoli Directory Server to Custom.

- ▶ Back in Global security, select *Lightweight Directory Access Protocol (LDAP) user registry* and click *OK* (Figure 11-23).

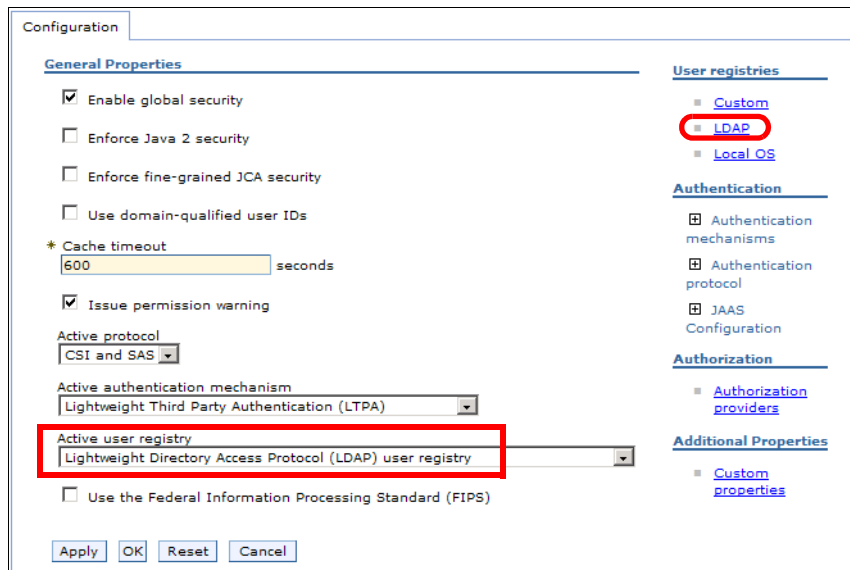


Figure 11-23 Global security using LDAP

- ▶ Save the configuration.

Tailor the LDAP transformation file

We have to tailor the LDAP transformation file to match the LDAP structure:

- ▶ Locate the transformation file:

```
<WID_HOME>\runtimes\bi_v6\ProcessChoreographer\Staff\LDAPTransformation.xsl
```

- ▶ Take a backup copy, then edit the file and make the changes highlighted in Example 11-1. We provide the LDAPTransformation.xsl file in:

```
SG247148\sampcode\ldap
```

Example 11-1 LDAP transformation file

```
.....
<!-- Begin global variables -->
<xsl:variable name="Threshold">20</xsl:variable>

<xsl:variable name="DefaultPersonClass">inetOrgPerson</xsl:variable>
<xsl:variable name="DefaultUserIdAttribute">uid</xsl:variable>
<xsl:variable name="DefaultMailAttribute">mail</xsl:variable>
<xsl:variable name="DefaultManagerAttribute">manager</xsl:variable>
```

```

<xsl:variable name="DefaultGroupClass">groupOfUniqueNames</xsl:variable>
<xsl:variable name="DefaultGroupClassMemberAttribute">uniquemember
</xsl:variable>

<xsl:variable name="DefaultRecursivity">yes</xsl:variable>

<!-- new variable -->
<xsl:variable name="MS_Domain">cn=groups,dc=ibm,dc=com</xsl:variable>
.....

<!-- Begin template GroupMembers and children -->
<xsl:template name="GroupMembers">
  <xsl:variable name="Group0">
    <xsl:value-of select="staff:parameter[@id='GroupName']"/>
    <xsl:if test="staff:parameter[@id='GroupName']!=''">
      <xsl:value-of select="$GS_GroupID"/>=
      <xsl:value-of select="staff:parameter[@id='GroupName']"/>,
      <xsl:value-of select="$MS_Domain"/>
    </xsl:if>
  </xsl:variable>
  <xsl:variable name="Group1">
    <xsl:value-of select="staff:parameter[@id='AlternativeGroupName1']"/>
    <xsl:if test="staff:parameter[@id='GroupName']!=''">
      <xsl:value-of select="$GS_GroupID"/>=
      <xsl:value-of select="staff:parameter[@id='AlternativeGroupName1']"/>,
      <xsl:value-of select="$MS_Domain"/>
    </xsl:if>
  </xsl:variable>
  <xsl:variable name="Group2">
    <xsl:value-of select="staff:parameter[@id='AlternativeGroupName2']"/>
    <xsl:if test="staff:parameter[@id='GroupName']!=''">
      <xsl:value-of select="$GS_GroupID"/>=
      <xsl:value-of select="staff:parameter[@id='AlternativeGroupName2']"/>,
      <xsl:value-of select="$MS_Domain"/>
    </xsl:if>
  </xsl:variable>
  <xsl:variable name="includesubgroups">
    <xsl:value-of select="staff:parameter[@id='IncludeSubgroups']"/>
  </xsl:variable>
.....

```

- ▶ These changes ensure that the LDAP query submitted by the server is in the form:

```
cn=groupname,cn=groups,dc=ibm,dc=com
```

Configuring the human tasks for LDAP

Open the human tasks (Review Order and Ship Order to Customer) and in the Properties view, Details tab (Figure 11-24), change the JNDI name to point to:

bpe/staff/sampleldapconfiguration

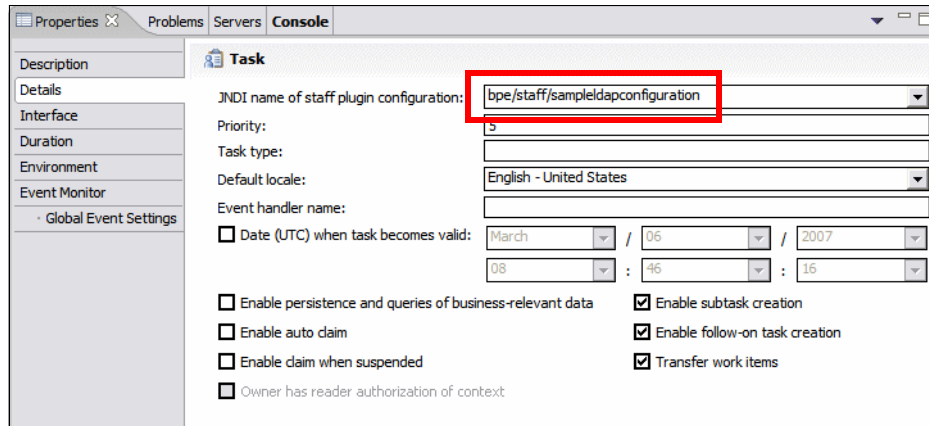


Figure 11-24 Human task configuration for LDAP

Testing LDAP security

Before you start the server with LDAP security, verify the server user ID and password in the server configuration (see “Specify security in Integration Developer” on page 311):

- ▶ Stop the server.
- ▶ Open the server configuration (see Figure 11-9 on page 311) and set the user ID and password to the same values as in Figure 11-21 on page 320.
- ▶ Start the server.

Tip: This sequence is important. Always stop the server with the old user ID, and start the server with the new user ID. To stop the server, the security credentials have to be provided.

Submit orders

You can now submit orders. Use the BPC Explorer (see “Testing human task security with the BPC Explorer” on page 314), and use the custom human task application (see “Using the human task Web application with security” on page 314).

Removing security

Once human task security is defined in the human tasks of the process, you cannot run the server without security. If you try to submit an order to the server running without security, you get error messages that work items cannot be created. You have to reconfigure the human tasks again and select *Everybody* as potential owner.


To run without security again, you have to change the server and the application.

- ▶ In the server administrative console, under global security, clear *Enable global security*. You can leave the other settings. Click *OK*, then save the configuration.
- ▶ Open the two human tasks (review order and ship order), select *Potential Owner*, and select *Everybody* in the Properties view.
- ▶ Stop the server.
- ▶ Open the server configuration and clear *Security is enabled on this server*. Save the configuration.
- ▶ Now you can restart the server.
- ▶ Restart the `ClipsAndTacksF1App` application.

Summary

In this chapter we described how to implement security in the Process Server so that we can control which users can process orders waiting for human interaction.

We showed two security implementations using a simple custom user registry and using an LDAP server for a production environment.



Developing and testing the business measures with the Monitor Toolkit

This chapter describes the steps performed in WebSphere Integration Developer Version 6.0.2 for creating business measures. We define all the relevant information about the BPEL process elements that are monitored by WebSphere Business Monitor Version 6.0.2 to verify the business goals.

This chapter also describes the configuration needed in the business measures model for providing the round trip back into the Business Modeler.

The topics outlined in this chapter are as follows:

- ▶ Importing the business measures created in WebSphere Modeler
- ▶ Defining the required events, triggers, KPI, metrics, and dimensions
- ▶ Creating the necessary configuration to be able to achieve the round trip
- ▶ Exporting the business measures to the Business Monitor test environment
- ▶ Configuring and running monitor models in the Monitor test environment
- ▶ Undeploying the model from the Monitor test environment

Note: Make sure that you have read Chapter 3, “WebSphere Business Monitor Version 6.0.2 architecture” on page 39.

Preparation in Modeler

In Chapter 9, “Defining KPIs and measures” on page 187 we defined the business measures definitions for the process. Now we have the opportunity to create the actual business measures model, called the *monitor model*, to measure the process in action, enabling us to measure the performance of the process, retrieving the information that flows through the Order Handling (Future 1) process.

Monitor Toolkit

WebSphere Business Monitor Development Toolkit provides the tools for creating monitor models, which can be transformed into executable code for WebSphere Business Monitor.

The WebSphere Monitor Development Toolkit runs as a plug-in into WebSphere Integration developer and includes the Monitor Model Editor and an integrated unit test environment (UTE).

The Monitor Model Editor is a visual editor for creating monitor models. The integrated test environment provided with the toolkit is used for testing the monitor models in WebSphere Integration developer.

The Monitor Toolkit must be installed in Integration Developer as described in “Installation of the Monitor Development Toolkit” on page 584.

Importing business measures into the Monitor Toolkit

This section describes about the importing the business measures artifacts created in the WebSphere Modeler in Chapter 9, “Defining KPIs and measures” on page 187.

Business Monitoring perspective

Before we start importing the business measures definition, we open WebSphere Integration Developer. We use the same work space that was used to develop the `ClipsAndTacksF1` application.

Open the Business Monitoring perspective by selecting *Window* → *Open Perspective* → *Others*, select *Show all*, and select *Business Monitoring* (Figure 12-1).

When prompted, click *OK* to enable the Monitor Model Editor capability.

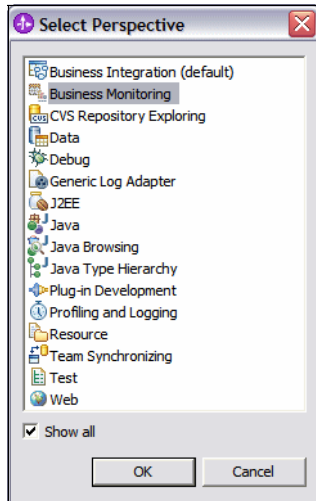


Figure 12-1 Business Monitoring perspective

Create a project for the business measures

We create a project to import the models from the Modeler:

- ▶ In the Project Explorer, select *New* → *Business Monitoring Project*. Enter *ClipsAndTacksF1mm* as project name and click *Finish*. (Figure 12-2).

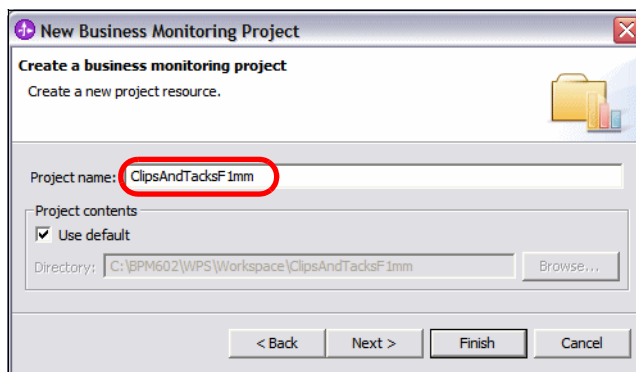


Figure 12-2 Create Business Monitoring Project

Import the business measures

Now let us import the business measure files:

- ▶ Select the *ClipsAndTacksF1mm* project and *Import*.
- ▶ Select *File system* and click *Next*.

- ▶ Specify the location of the file system, where the exported files from the Modeler for business measures were exported:

C:\SG247148\sampcode\model\export

- ▶ Select the two .svg files and the .mm file. Make sure the Into Folder contains ClipsAndTacksF1mm. Click *Finish* (Figure 12-3).

Note that errors show up in the Project Explorer for the project.

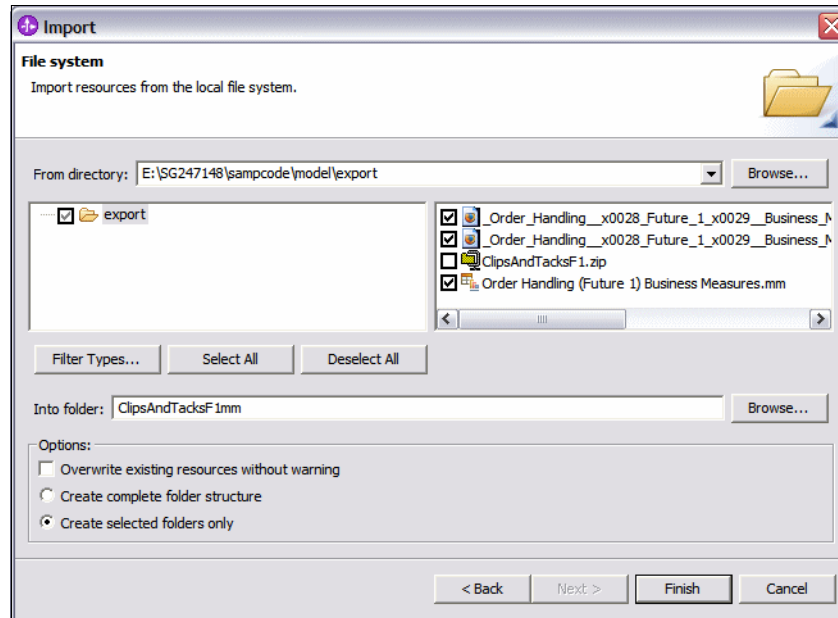


Figure 12-3 Import the business measures files

Studying the imported business measures

To find out what the business analysts want to monitor, we open the imported business measures and study the definitions:

- ▶ Open the Order Handling (Future 1) Business Measures.
- ▶ The Monitor Details Model tab shows the duration and processing time measurements that the Monitor should feed back to the Modeler in the round-trip scenario (Figure 12-4).

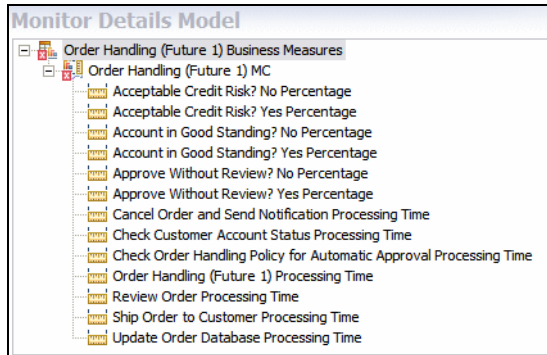


Figure 12-4 Monitor Details Model: imported

- ▶ The Data Mart Model tab shows the Location dimension, a number of facts, and a number of measures, such as Order Count and Order Price Total (Figure 12-5).

Data Mart Model

Dimensions
Add dimensions and their hierarchical attributes. Each attribute level enables you to aggregate the levels that are underneath it.

Cube / Dimension	Dimension Attribute	Source	Is Key
Order Handling (Future 1) MC Cube		Order Handling (Future 1) MC	
Location			

Facts
A fact is required for each metric, counter, and stopwatch that is not a dimension attribute. Facts are the basis for any measures you create.

Cube / Fact Table	Fact Attribute	Source
Order Handling (Future 1) MC Cube		Order Handling (Future 1) MC
Order_Handling__x0028_Future_1...	Account_in_Good_Standing_...	Account in Good Standing? Yes Percentage
	Account_in_Good_Standing_...	Account in Good Standing? No Percentage
	Review_Order_Processing_T...	Review Order Processing Time
	Acceptable_Credit_Risk_x00...	Acceptable Credit Risk? Yes Percentage
	Acceptable_Credit_Risk_x00...	Acceptable Credit Risk? No Percentage

Measures
Add measures, which are calculations performed on facts to collect and combine them into a single value.

Cube	Measure	Source	Aggregation Function
Order Handling (Future 1) MC Cube		Order Handling (Future 1) MC	
	Order Count		Count
	Shipped Order Count		Count
	Declined Order Count		Count
	Order Price Total		Sum
	Order Price Average		Average
	Average Account in Good Standing? ...	Account_in_Good_Standing_x003...	Average

Figure 12-5 Data Mart Model: imported

- ▶ The KPI Model tab shows the KPIs defined in the Modeler (Figure 12-6).

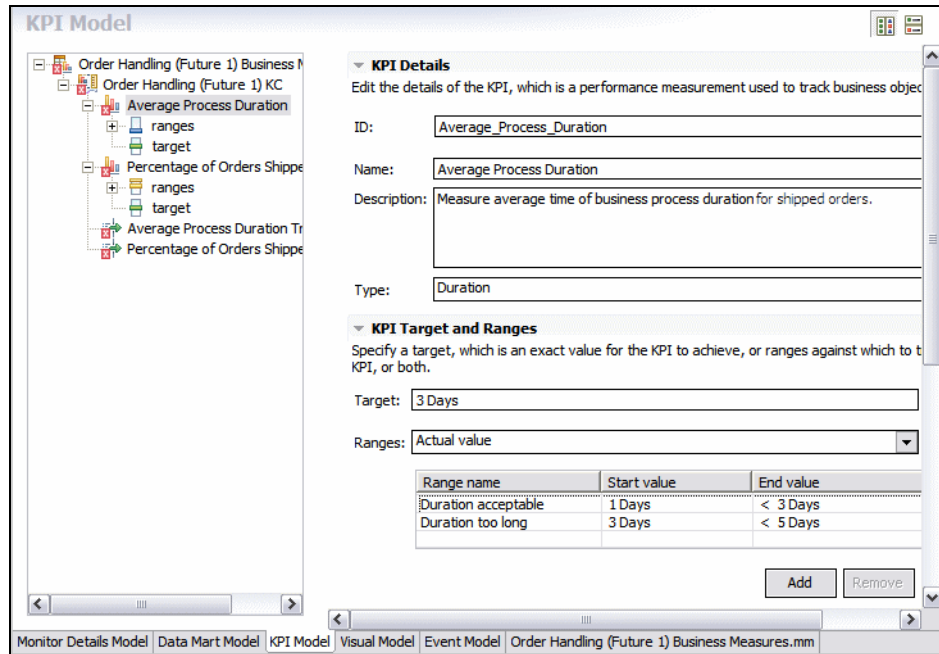


Figure 12-6 KPI Model: imported

- The Visual Model tab shows the process diagram (Figure 12-7).

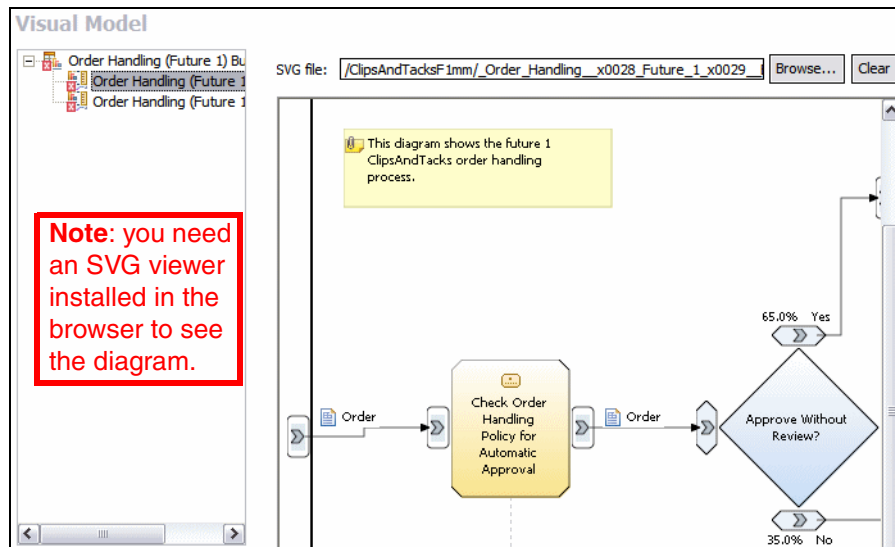


Figure 12-7 Visual Model: imported

- ▶ The Event Model tab is empty.
- ▶ The Order Handling (Future 1) Business Measures.mm tab shows the XML file that contains all the definitions (Figure 12-8).

```

<?xml version="1.0" encoding="UTF-8"?>
<mm:monitor xmlns:mm="http://www.ibm.com/xmlns/prod/websphere/monitoring/6.0.2/mm" xm
<monitorDetailsModel displayName="Order Handling (Future 1) Business Measures" id="
  <monitoringContext displayName="Order Handling (Future 1) MC" id="Order_Handling_
    <metric displayName="Account in Good Standing? Yes Percentage" id="Account_in_G
      <defaultValue>
        <singleValue expression="0"/>
      </defaultValue>
    </metric>
    <metric displayName="Account in Good Standing? No Percentage" id="Account_in_Go
      <defaultValue>
        <singleValue expression="0"/>
      </defaultValue>
    </metric>
    <metric displayName="Review Order Processing Time" id="Review_Order_Processing_
      <defaultValue>
        <singleValue expression="duration('PT0S')"/>
      </defaultValue>
    </metric>
    <metric displayName="Acceptable Credit Risk? Yes Percentage" id="Acceptable_Cre
  </monitoringContext>
</monitorDetailsModel>

```

Figure 12-8 Monitor Model XML file: imported

- ▶ Close the imported monitor model.

Creating the monitor model

The imported measures from Modeler will be used as a reference to IT developers for the business measures that have to be developed.

We do not use the imported model for the actual creation of the business measures.

Note: The main reason to develop a new monitor model is the fact that no events are available in the imported model. Generating the events into the existing monitor model deletes all the existing definitions. The only way to get the events into the imported model is to merge the .mm XML files from a generated monitor model into the imported monitor model. However, this is very error prone and we decided to use a new monitor model.

To create the business measures, we perform the following steps:

- ▶ Generate CEI events for BPEL elements.
- ▶ Generate Monitor events.
- ▶ Generate the Monitor model.
- ▶ Create the respective business measures (metric, KPI, dimensions, alerts).

Generating CEI events for BPEL elements

This section describes about setting the properties for the BPEL components to generate the CEI events.

Change to the Business Integration perspective and open the Order Handling (Future 1) process:

- ▶ Select the *OrderHandling(Future1)Receive* element, in the process. In the select Properties view, select *Event Monitor* (Figure 12-9).

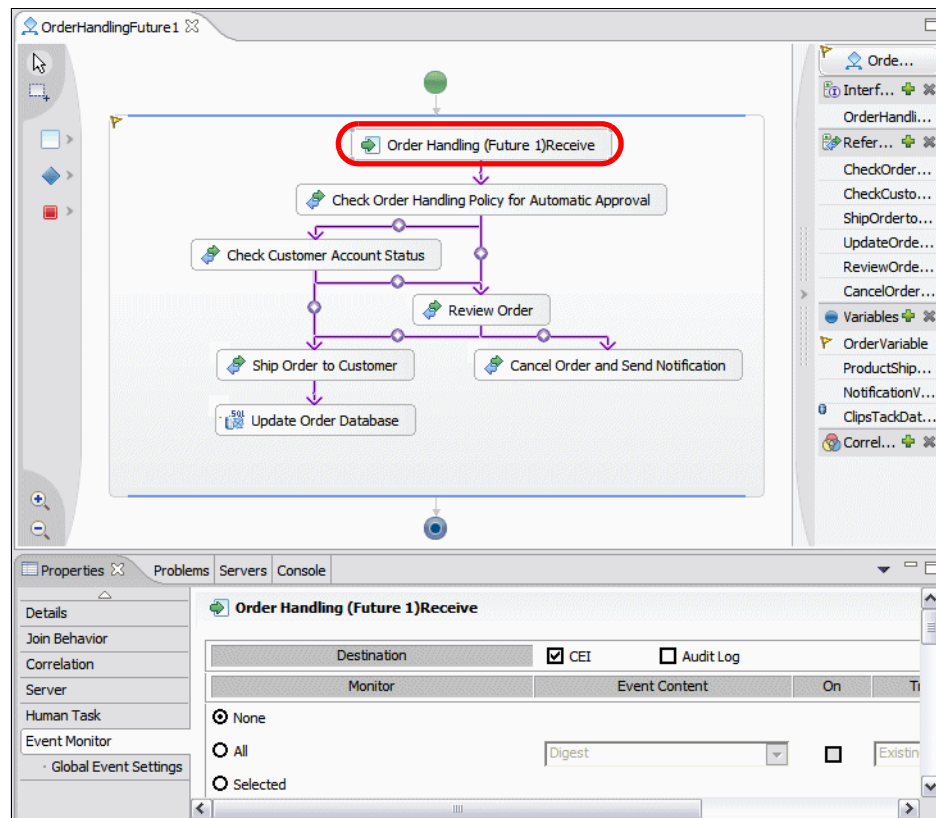


Figure 12-9 Event Monitor for an activity

- ▶ In the Event Monitor Tab, Select *All* under Monitor and *Full* under Event Context (Figure 12-10).

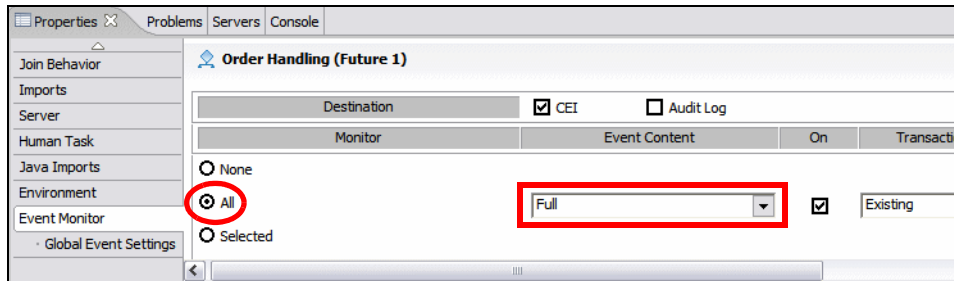


Figure 12-10 Specify that all events are created with full content

- ▶ A yellow flag is set on the OrderHandling (Future 1) Receive element in the process (Figure 12-11).

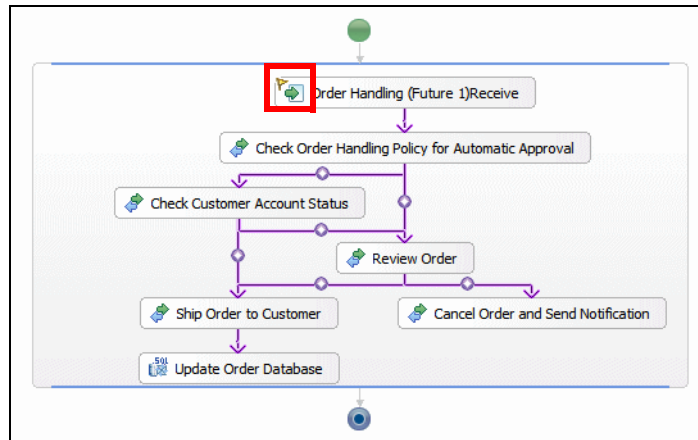


Figure 12-11 Event flag in the BPEL process

Continue configuring the CEI events for the all the elements inside the BPEL process:

- ▶ Check Order Handling Policy for Automatic Approval
- ▶ Check Customer Account Status
- ▶ Review Order
- ▶ Ship Order to Customer
- ▶ Cancel Order and Send Notification
- ▶ Update Order Database
- ▶ OrderHandlingFuture1_Flow (select the process flow rectangle)
- ▶ Order Handling (Future 1) (the process itself, click outside of the flow).

Save the process, and all the elements and the process are flagged (Figure 12-12).

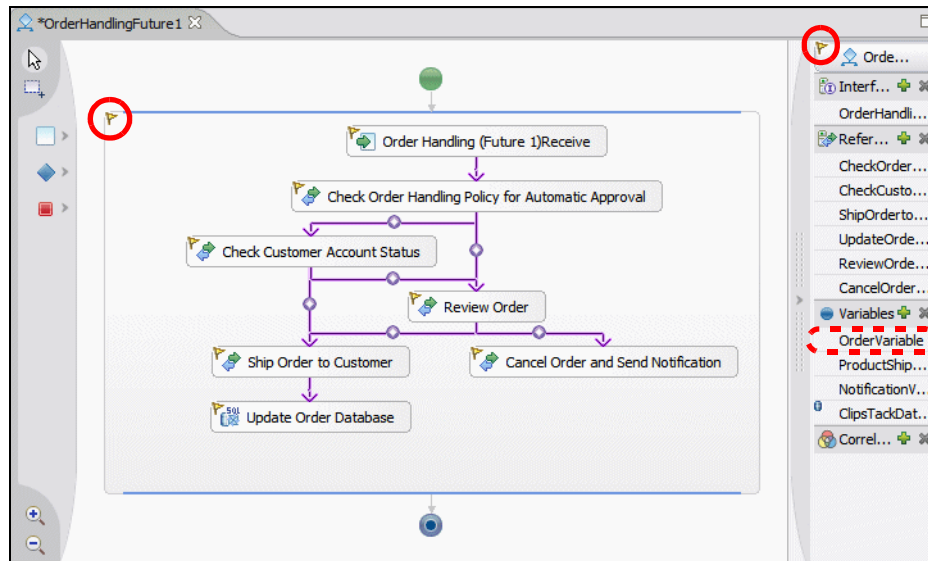


Figure 12-12 All process elements events are generated

Events for variables

To retrieve data from an order (order price, customer city) we have to configure the CEI event for the `OrderVariable` business object:

- ▶ Select `OrderVariable` under Variables (Figure 12-12), and configure the CEI event with `All` and `Full` (Figure 12-13).

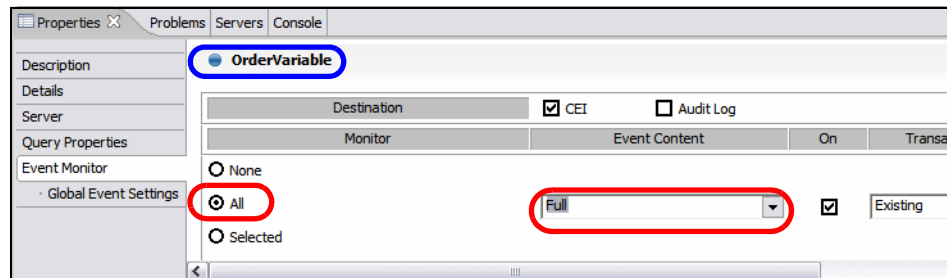


Figure 12-13 CEI for Order Variable

- ▶ The `OrderVariable` is flagged.
- ▶ Save the process.

Generating monitoring events

Once the CEI events are defined, we generate the monitoring events:

- ▶ Select the Order Handling (Future 1) process in the Business Integration view, select *Monitor Tools*, and select *Generate Event Definitions*.

Be patient... this operation takes a while.

- ▶ A confirmation window opens (Figure 12-14), Click *OK*.

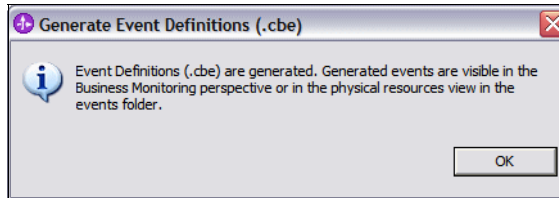


Figure 12-14 Created .cbe files

- ▶ The events are visible in the Business Monitoring perspective Project Explorer (Figure 12-15).

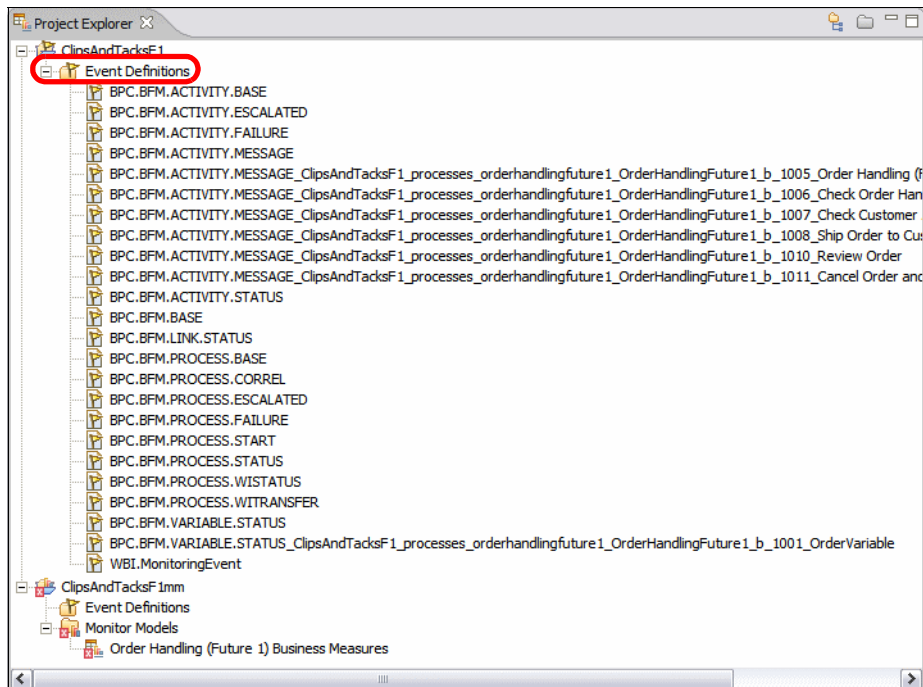


Figure 12-15 Generated monitoring events

Generating the monitor model

Next we generate the monitor model:

- ▶ Select the Order Handling (Future 1) process in the Business Integration view, and select *Monitor Tools* and *Generate Monitor Model*.
- ▶ Click *New project* for the business measures (we are not using the project used for importing). Type *ClipsAndTacksF1BMP* as the project name and click *Finish* (Figure 12-16).

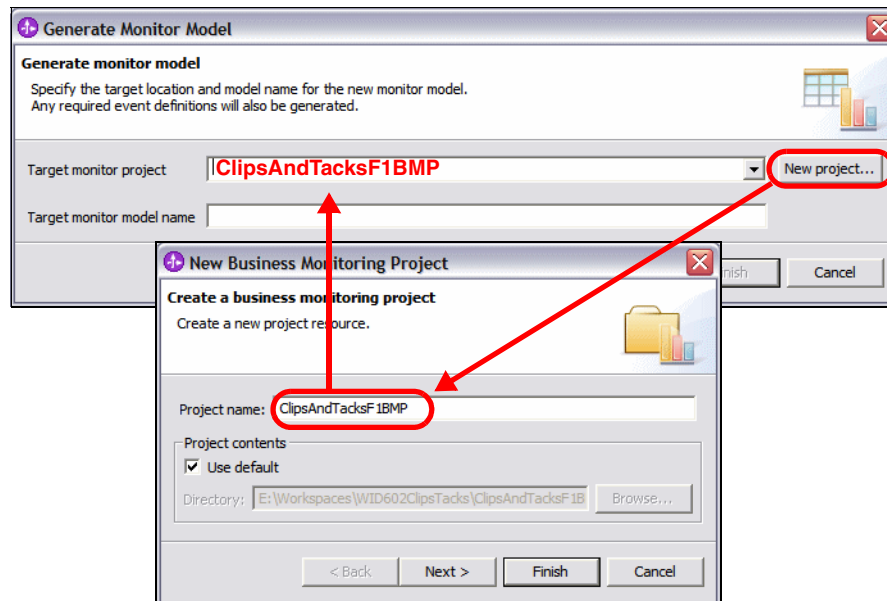


Figure 12-16 Generate the monitor model: Business monitoring project

- ▶ Type *ClipsAndTacksF1BMP* as the monitor model name and click *Next* (Figure 12-17).

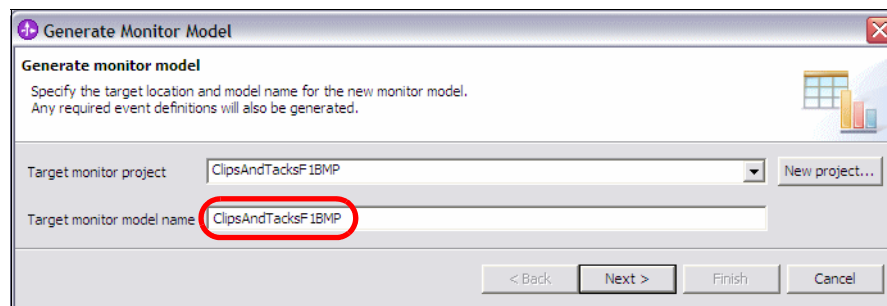


Figure 12-17 Monitor model name

- ▶ On the next panel, select all the events we are interested in monitoring. Select all the elements of the process and the `OrderVariable`, then click *Finish* (Figure 12-18). When you select an element in the left pane, you can see the events of that element in the right pane (selected by default).

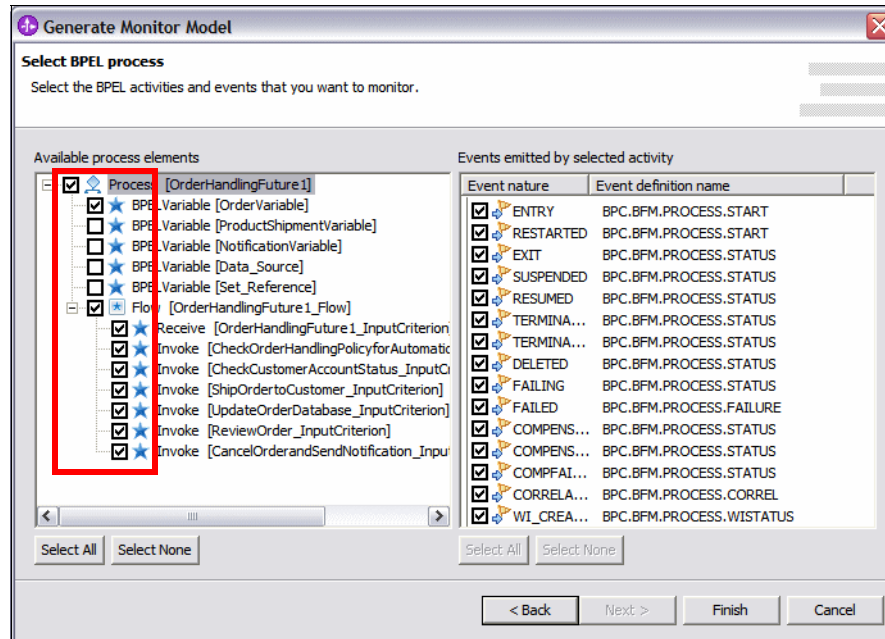


Figure 12-18 Events to be monitored

- ▶ When prompted, click *Yes* to change to Business Monitoring perspective.

Monitor Model Editor

The Monitor Model Editor opens (Figure 12-19):

- ▶ The Monitor Details Model tab shows all the monitoring contexts and their events.
- ▶ The Data Mart Model tab shows the cube dimensions, facts, and measures.
- ▶ The KPI Model tab shows the KPIs (none at this point).
- ▶ The Visual Model tab shows the diagram (if we copy the SVG files).
- ▶ The Event Model tab shows the events (this is where the imported model is empty).
- ▶ The `ClipsAndTacksF1BMP.mm` tab shows the underlying XML file.

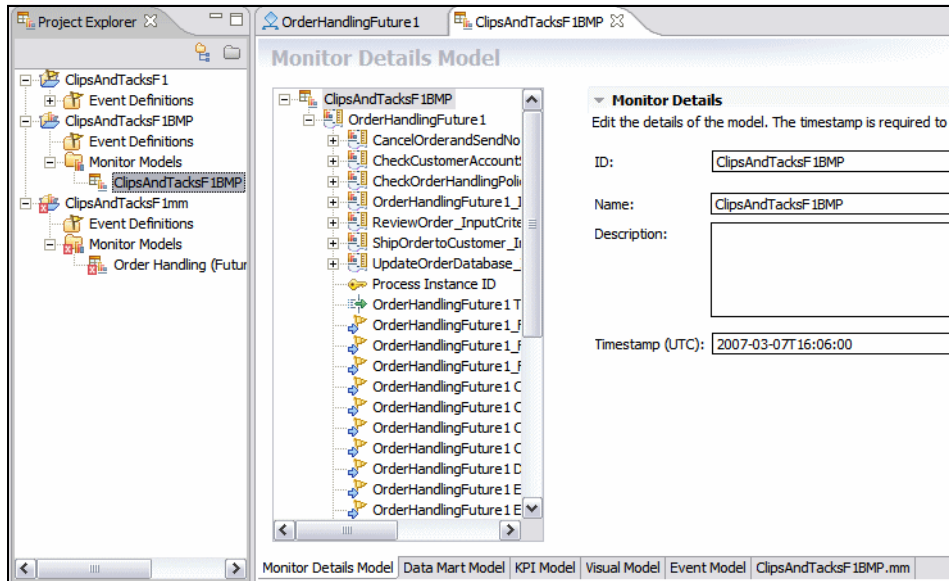


Figure 12-19 Business monitoring model generated

There is a monitoring context for the process (OrderHandlingFuture1) and for each activity in the process (CancelOrderandSendNotification, and so forth). A process instance ID, a termination trigger, and the events of the process are listed at the bottom.

When you expand the monitoring context of an activity, you can see two instance IDs, a termination trigger, and a number of events (Figure 12-20).

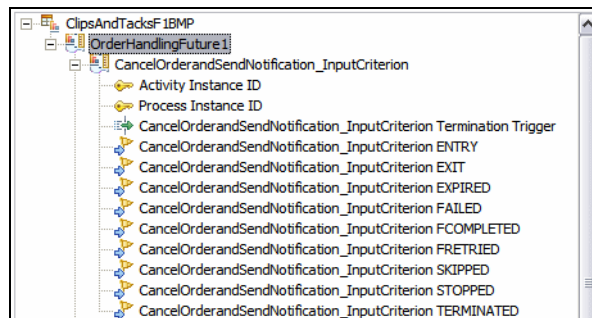


Figure 12-20 Monitoring context of an activity

Creating metrics and measures

In this section we create the metrics and measures that we want to monitor.

Order Count

This measure counts the number of orders that are submitted.

Note: All metrics, measures, and KPIs defined in this chapter can be achieved in a number of different ways. The approach used in this chapter illustrates the different options and capabilities available in the Monitor Toolkit.

Create a trigger

To count the orders, we use a trigger when a process instance starts:

- ▶ In the Monitor Details Model tab, select *ClipsAndTacksF1BMP* → *OrderHandlingFuture1* and *New* → *Trigger* (Figure 12-21).

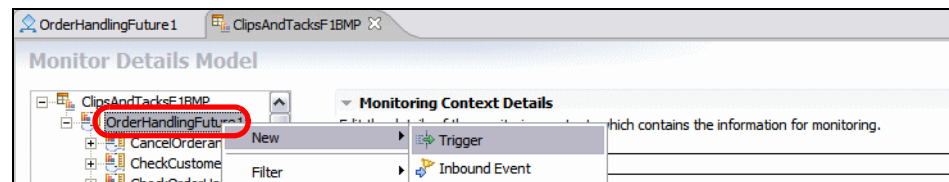
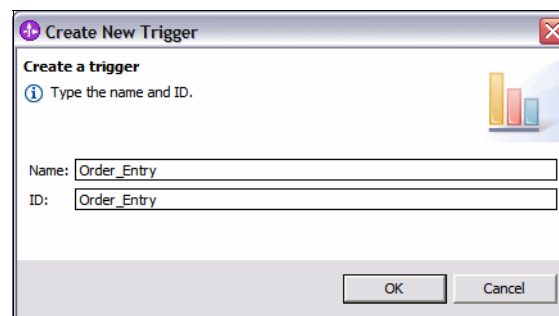


Figure 12-21 Order entry trigger

- ▶ Name the trigger *Order_Entry* (Figure 12-22).



The element names can be anything. The ID is generated from the name and cannot include spaces.

Figure 12-22 Trigger name

- ▶ The trigger is added to the *OrderHandlingFuture1* monitoring context.
- ▶ In the trigger details, scroll down to *Trigger Sources* and click *Add*.

- ▶ Select *Other source type* and select *OrderHandlingFuture1Entry* (Figure 12-23).

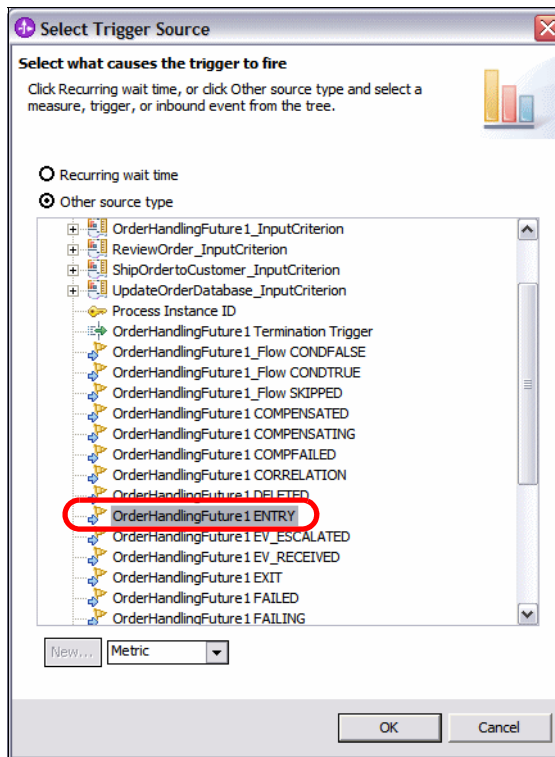


Figure 12-23 Select the event of a trigger

- ▶ The source event is added to Trigger Sources (Figure 12-24).

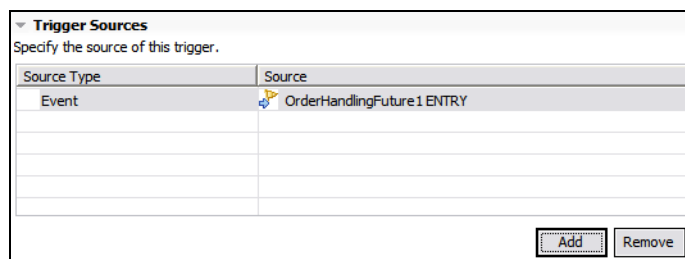


Figure 12-24 Trigger source created

Create a counter

We use the order entry trigger to count the orders using a counter:

- ▶ Select *OrderHandlingFuture1* and *New* → *Counter*. Name the counter *Order_Count*.
- ▶ Scroll down to *Counter Controls* and click *Add*.
- ▶ Select the *Order_Entry* trigger source and click *OK*.
- ▶ The default action of *Add One* is correct for our purpose.
- ▶ The Details view of the counter is shown in Figure 12-25 (after adding the trigger).

The screenshot shows the 'Counter Details' configuration window. The 'ID' and 'Name' fields are both set to 'Order_Count'. There is a description field and a checkbox for 'This counter can be used for sorting'. Below this is the 'Counter Controls' section, which contains a table with two columns: 'Trigger / Inbound Event' and 'Resulting Action'. The table has one row with 'Order_Entry' in the first column and 'Add One' in the second column. A red arrow points from the 'Add One' cell to the 'Add' button at the bottom right of the window. The 'Add' button is circled in red.

Trigger / Inbound Event	Resulting Action
Order_Entry	Add One

Figure 12-25 *Order_Count* detail

Monitoring flow

The Monitoring Flow view shows the flow of how the counter is manipulated (Figure 12-26).

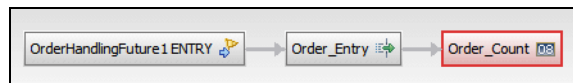


Figure 12-26 *Monitoring flow*

Creating the order count measure

Once the metric is created, we create the measure. The metric gives the value for a particular instance, whereas the measure calculates the value across many instances.

Select the *Data Mart Model* tab (where measures are defined):

- ▶ Scroll down to Measures, select OrderHandlingFuture1Cube, and click *Add Measure* (Figure 12-27).

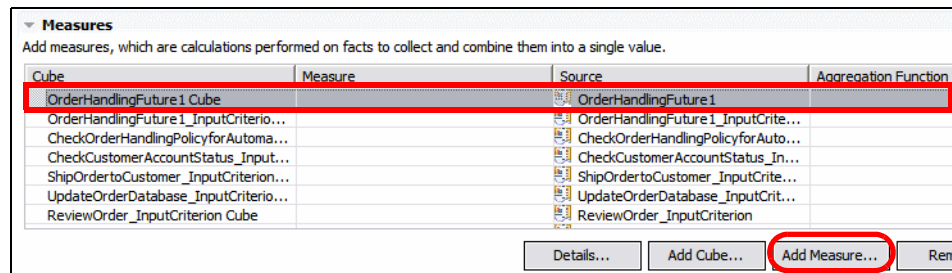


Figure 12-27 Create measure

- ▶ Name the measure Order Count and click *OK* (Figure 12-28).

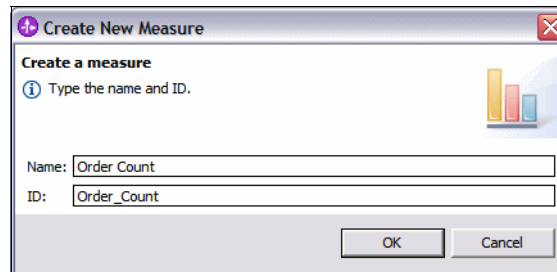



Figure 12-28 Measure name

- ▶ Click  in the Source column for the Order Count measure (Figure 12-29).

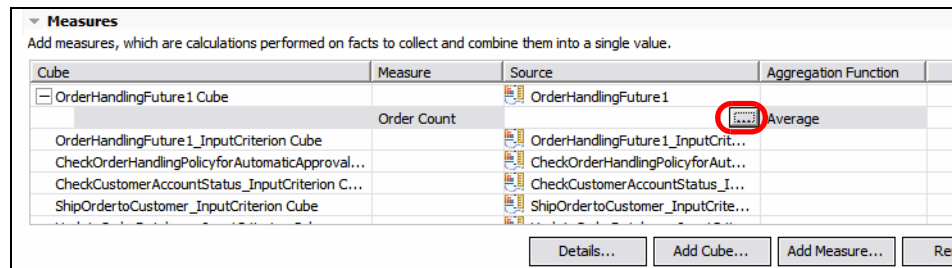


Figure 12-29 Source for the measure

- ▶ Select the *Order_Count* fact and click *OK* (Figure 12-30). Every metric becomes a fact in the Data Mart Model tab.

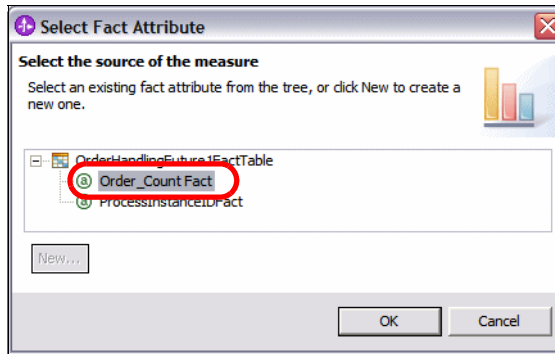


Figure 12-30 Select the fact of the metric

- ▶ For the Aggregation Function, select *Sum* (Figure 12-31).

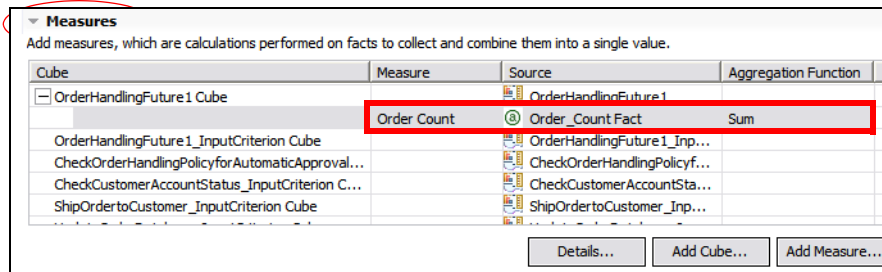


Figure 12-31 Aggregation function for Order Count measure

We have created the Order Count measure.

Shipped Order Count

This measure counts the number of orders that are shipped. In our business process, an order can be shipped or cancelled.

We count the total number of orders that are approved automatically and shipped, and orders that are approved by an order manager.

Create the Shipped Order Count measure with the values as stated in Table 12-1.

Table 12-1 Shipped Order Count Measure details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: OrderHandlingFuture1 → ShipOrdertoCustomer_InputCriterion	Name: ShipOrder_Exit
Associate event to trigger	Trigger Source: ShipOrdertoCustomer_InputCriterion	Select: ShipOrdertoCustomer_Input Criterion EXIT
Create counter	Counter Location: ShipOrdertoCustomer_InputCriterion	Name: Shipped Order Count
Associate trigger to counter	Counter Controls: ShipOrdertoCustomer_InputCriterion	Trigger: ShippedOrder_Exit Resulting action: Add One
Ass measures	Data Mart Model → Measures: ShipOrdertoCustomer_InputCriterion Cube	Name: Shipped Order Count Source: Shipped Order Count Fact Aggregation Function: Sum

The resulting measure is shown in Figure 12-32.

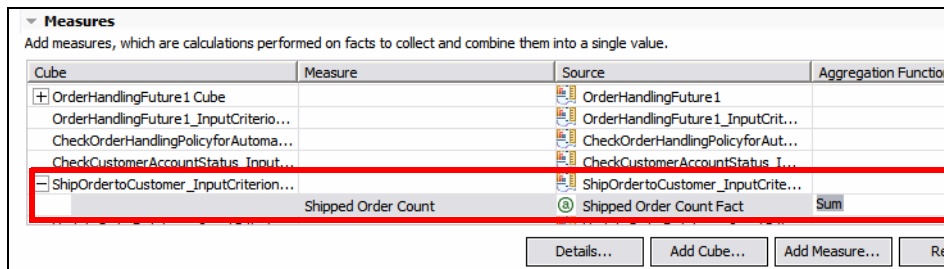


Figure 12-32 Shipped Order Count measure

Declined Order Count

This measure counts the number of orders that were declined by the order manager. Orders are reviewed by the order manager when the order price is more than \$750 or the credit check is not acceptable.

Create the Declined Order Count measure with the values as stated in Table 12-2.

Table 12-2 Declined Order Count measure details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: OrderHandlingFuture1 → CancelOrderandSendNotification_InputCriterion	Name: CancelOrder_Entry
Associate event to trigger	Trigger Source: CancelOrderandSendNotification_InputCriterion	Select: CancelOrderandSendNotification_InputCriterion ENTRY
Create counter	Counter Location: OrderHandlingFuture1 → CancelOrderandSendNotification_InputCriterion	Name: Declined Order Count
Associate trigger to counter	Counter Controls: CancelOrderandSendNotification_InputCriterion	Trigger: CancelOrder_Entry Resulting action: Add One
Add measure	Data Mart Model → Measures: CancelOrderandSendNotification_InputCriterion Cube	Name: Declined Order Count Source: Declined Order Count Fact Aggregation Function: Sum

The measure is shown in Figure 12-33.

The screenshot shows a table titled 'Measures' with the following data:

Cube	Measure	Source	Aggregation Function
CheckCustomerAccountStatus_Input...		CheckCustomerAccountStatus_I...	
ShipOrdertoCustomer_InputCriterion...		ShipOrdertoCustomer_InputCrite...	
UpdateOrderDatabase_InputCriterio...		UpdateOrderDatabase_InputCrit...	
ReviewOrder_InputCriterion Cube		ReviewOrder_InputCriterion	
CancelOrderandSendNotification_In...	Declined Order Count	Declined Order Count Fact	Sum

A red box highlights the row for 'CancelOrderandSendNotification_In...' and its corresponding source and aggregation function.

Figure 12-33 Aggregation function for declined orders


Order Price Total

This measure aggregates the price of all orders submitted. The order price is available in the `OrderVariable`, therefore we capture the price at the value change events of the `OrderVariable`. Note that the metric only records the value of the last change event (and the price never changes).

Create the Order Price Total measure with the values as stated in Table 12-3.

Table 12-3 Order Price Total measure details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: OrderHandlingFuture1	Name: OrderVariable_Changed
Associate event to trigger	Trigger Source: OrderHandlingFuture1	Select: OrderVariable_CHANGED
Create metric	Metric Location: OrderHandlingFuture1	Name: Order Price Type: Decimal Default Value: 0.00
Associate trigger to metric	Trigger Location: OrderHandlingFuture1	Metric Value Maps: Click <i>Add</i> . Trigger: OrderVariable_Changed Expression: OrderVariable_CHANGED/extendedData/variableData/TotalPrice ^a
Add measure	Data Mart Model → Measures: OrderHandlingFuture1 Cube	Name: Order Price Total Source: Order Price Fact Aggregation Function: Sum

a. Click  in the Expression column, then use **Ctrl+Space** and select *ClipsAndTacksF1BMP* → *OrderHandlingFuture1* → *OrderVariable_CHANGED* → *extendedData* → *variableData* → *TotalPrice*.

The creation of the metric value expression is shown in Figure 12-34.

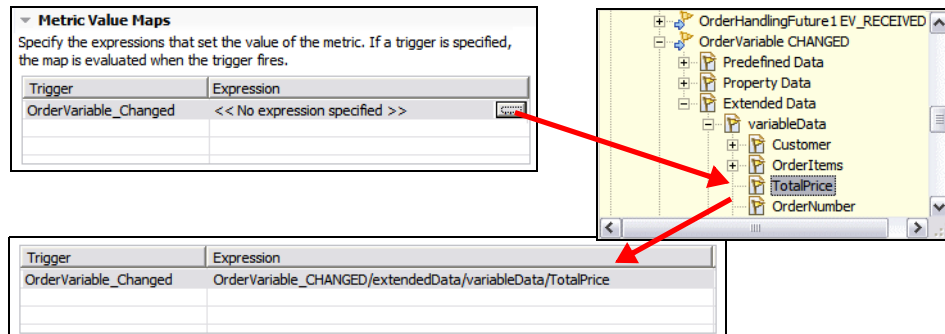


Figure 12-34 Order Price Total metric triggers and expression

The measure is shown in Figure 12-35.

Cube	Measure	Source	Aggregation Function
OrderHandlingFuture1 Cube	Order Count	Order_Count Fact	Sum
	Order Price Total	Order Price Fact	Sum
	OrderHandlingFuture1_InputCriteria...	OrderHandlingFuture1_InputCrit...	
	CheckOrderHandlingPolicyforAutoma...	CheckOrderHandlingPolicyforAut...	
CheckCustomerAccountStatus_Input...	CheckCustomerAccountStatus_I...		

Figure 12-35 Aggregation function for the Order Price Total

Order Price Average

This measure calculates the average order price for all orders submitted. We already have the underlying metric, therefore we can define the measure immediately as shown in Table 12-4.

Table 12-4 Order Price Average measure details

Activity	Description / Location	Name / Value
Add measure	Data Mart Model → Measures: OrderHandlingFuture1 Cube	Name: Order Price Average Source: Order Price Fact Aggregation Function: Average

Figure 12-36 shows the measure.

Cube	Measure	Source	Aggregation Function
OrderHandlingFuture1 Cube	Order Count	Order_Count Fact	Sum
	Order Price Total	Order Price Fact	Sum
	Order Price Average	Order Price Fact	Average
	OrderHandlingFuture1_InputCriteria...	OrderHandlingFuture1_InputCrit...	
CheckOrderHandlingPolicyforAutoma...	CheckOrderHandlingPolicyforAut...		

Figure 12-36 Aggregation function for Average Order Price

Creating a dimension

Dimensions give the users the ability to view the numeric data (for example, the measures) with respect to the configured dimensions. We configure a *Location* dimension that is based on the customers country and city. This enables us to see numeric data, such as order counts and total/average price by country or city.

Location dimensions by country and city

The first step is to create two metrics for Order Country and Order City, and then create the Location dimension based on these two metrics.

The customer's country and city are stored in the `OrderVariable`.

Order City and Order Country metrics

We extract the customer's city and country from the `OrderVariable` as shown in Table 12-5.

Table 12-5 Order City and Order Country metrics details


Activity	Description / Location	Name / Value
Create metric	Metric Location: OrderHandlingFuture1	Name: Order City Type: String Length: 20 Default Value: " (empty string—two single quotes)
Associate trigger to metric	Trigger Location: OrderHandlingFuture1	Metric Value Maps: Trigger: OrderVariable_Changed Expression: OrderVariable_CHANGED/extendedData/variableData/Customer/City (Click  in the Expression column, then use Ctrl+Space to select the city)
Create metric	Metric Location: OrderHandlingFuture1	Name: Order Country Type: String Length: 20 Default Value: " (empty string—two single quotes)
Associate trigger to metric	Trigger Location: OrderHandlingFuture1	Metric Value Maps: Trigger: OrderVariable_Changed Expression: OrderVariable_CHANGED/extendedData/variableData/Customer/Country

Figure 12-37 demonstrates the creation of the Order City metric.

The screenshot shows the 'Metric Details' configuration window. The 'ID' field is 'Order_City'. The 'Name' field is 'Order City'. The 'Type' dropdown is set to 'String'. The 'Maximum String Length' is set to '20'. The 'Default Value' is set to '*'. The 'Metric Value Maps' table is highlighted with a red box and contains the following data:

Trigger	Expression
OrderVariable_Changed	OrderVariable_CHANGED/extendedData/variableData/Customer/City

Figure 12-37 Order City metric details

Create the Location dimension


Once the Order City and Order Country metrics are created we can add these metrics to a dimension:

- ▶ In the Data Mart Model tab, Dimensions section, select *OrderhandlingFuture1Cube* and click *Add Dimensions* (Figure 12-38).

The screenshot shows the 'Data Mart Model' window. The 'Dimensions' section is expanded, and the 'OrderHandlingFuture1Cube' is selected. The 'Add Dimension...' button is highlighted with a red box.

Cube / Dimension	Dimension Attribute	Source	Is Key
OrderHandlingFuture1Cube		OrderHandlingFuture1	
OrderHandlingFuture1_InputCriteria...		OrderHandlingFuture1_InputCrit...	
CheckOrderHandlingPolicyforAutoma...		CheckOrderHandlingPolicyforAut...	
CheckCustomerAccountStatus_Input...		CheckCustomerAccountStatus_I...	
ShipOrdertoCustomer_InputCriteria...		ShipOrdertoCustomer_InputCrit...	
UpdateOrderDatabase_InputCriteria...		UpdateOrderDatabase_InputCrit...	

Figure 12-38 Adding a dimension

- ▶ Specify the dimension name as Location.
- ▶ In the Dimensions section, select *Location* and click *Add Attribute*.
- ▶ Specify the attribute name as Country.
- ▶ Click  in the Source column and select the Order Country metric. Leave Key selected (Figure 12-39).

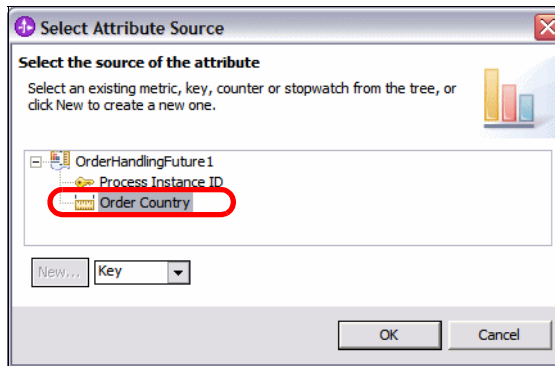


Figure 12-39 Order Country metric

- ▶ Create another attribute named City and select the Order City metric.

Note: The sequence of the attributes while creating a dimension is important. In this scenario we will drill down the orders first by country, and then by city, and not vice versa.

- ▶ The Dimensions section is shown in Figure 12-40.

Cube / Dimension	Dimension Attribute	Source	Is Key
OrderHandlingFuture1.Cube		OrderHandlingFuture1	
Location			
	Country	Order Country	<input checked="" type="checkbox"/>
	City	Order City	<input checked="" type="checkbox"/>
Shipped Flag			
Start Time			
OrderHandlingFuture1_InputCriteria...		OrderHandlingFuture1_InputCrite...	

Figure 12-40 Dimensions section

Creating KPIs

This section describes the configuration steps needed to configure the KPIs for the Order Handling process.

Average order fulfillment time for shipped orders

The key performance indicator (KPI) Average Shipped Order Fulfillment Time is the average time taken to process the orders that are approved and shipped. To implement this KPI, we need the duration between start time of the process and the end time of process, but only for shipped orders. We use a stopwatch to measure the time and ensure that we only measure shipped orders.

Create a stopwatch for the order processing time

We already created an Order_Entry trigger, now we add an Order_Exit trigger. Then we create a stopwatch to measure the time between these two events, which is the total duration of an order in the process. Later we use a dimension as a filter to get us the duration of shipped orders.

Table 12-6 shows how the stopwatch is created.

Table 12-6 Stop watch for Orders process time

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: OrderHandlingFuture1	Name: Order_Exit
Associate event to trigger	Trigger Source: OrderHandlingFuture1	Select: OrderHandlingFuture1 EXIT
Create Stopwatch	Stopwatch location: OrderHandlingFuture1	Name: Order Process Time
Associate trigger to stopwatch	Trigger Location: OrderHandlingFuture1	Stopwatch Controls: Click <i>Add</i> twice Trigger 1: Order_Entry Resulting Action: Start Trigger 2: Order_Exit Resulting Action: Stop

Figure 12-41 shows the Order Process Time stopwatch.

Figure 12-41 Order Process Time stopwatch with start and stop triggers

Shipped Flag metric

We determined the Order Process Time for both shipped orders and cancelled orders. We filter the shipped orders by using a dimension.

Create the Shipped Flag metric with the values of Table 12-7.

Table 12-7 Order Shipped Flag metric details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: OrderHandlingFuture1	Name: Order_Shipped
Associate event to trigger	Trigger Source: ShipOrdertoCustomer_InputCriterion	Select: ShipOrdertoCustomer_InputCriterion EXIT
Create metric	Metric Location: ClipsAndTacksF1BMP → OrderHandlingFuture1	Name: Shipped Flag Type: String Length: 12 Default Value: 'NOT_SHIPPED'
Associate trigger to metric	Trigger Location: OrderHandlingFuture1	Metric Value Maps: Trigger: Orders_Shipped Expression: 'SHIPPED'

Figure 12-42 shows the Shipped Flag metric.

The screenshot shows the 'Metric Details' configuration window. The 'ID' field is 'Shipped_Flag', 'Name' is 'Shipped Flag', and 'Type' is 'String'. The 'Maximum String Length' is set to 12. The 'Default Value' is 'NOT_SHIPPED'. A table under 'Metric Value Maps' shows a trigger 'Order_Shipped' with the expression 'SHIPPED'.

Trigger	Expression
Order_Shipped	'SHIPPED'

Figure 12-42 Shipped Flag metric details

Shipped Flag dimension

In the Data Mart Model create a dimension under the OrderHandlingFuture1 Cube (Figure 12-43):

- ▶ For the name type Shipped Flag.
- ▶ Add an attribute named Shipped with the Shipped Flag metric as source.

Figure 12-43 shows the Dimensions section with the Shipped Flag.

The screenshot shows the 'Dimensions' section with a table listing dimensions. The 'Shipped Flag' dimension is highlighted with a red box. It is under the 'OrderHandlingFuture1 Cube' and has a 'Shipped' attribute with 'Shipped Flag' as the source.

Cube / Dimension	Dimension Attribute	Source	Is Key
OrderHandlingFuture1 Cube		OrderHandlingFuture1	
Location			
Shipped Flag	Shipped	Shipped Flag	<input checked="" type="checkbox"/>
OrderHandlingFuture1_InputCriterio...		OrderHandlingFuture1_InputCrit...	
CheckOrderHandlingPolicyforAutoma...		CheckOrderHandlingPolicyforAut...	

Figure 12-43 Dimensions section with Shipped Flag

Start Time dimension

We want to review the order fulfillment time for every week, therefore the average order fulfillment time has to be a dimension of time. Let us create a metric with the start time and add it as a dimension.

Create the Start Time metric with the values of Table 12-8.

Table 12-8 Start Time metric and dimension

Activity	Description / Location	Name / Value
Create metric	Metric Location: OrderHandlingFuture1	Name: Start Time Type: DateTime Default Value: Click <i>Edit</i> and select the current date
Associate trigger to metric	Trigger Location: OrderHandlingFuture1	Metric Value Maps: Trigger: Order_Entry Expression: OrderHandlingFuture1_ENTRY/predefined Data/creationTime
Create dimension	Cube: OrderHandlingFuture1	Name: Start Time Attribute: Time Source: Start Time

Figure 12-44 shows the Start Time dimension.

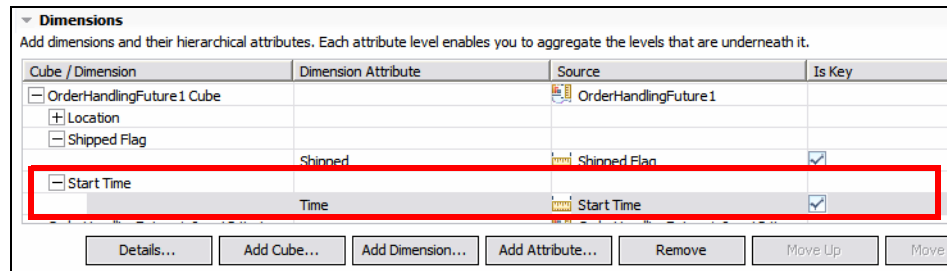


Figure 12-44 Dimensions section with Start Time

Order Fulfillment Time measure

Create the Order Fulfillment Time measure as the average of the order process time (Table 12-9).

Table 12-9 Order fulfillment time Measure details

Activity	Description / Location	Name / Value
Measures	Data Mart Model → Measures: OrderHandlingFuture1 Cube	Name: Order Fulfillment Time Source: Order Process Time fact Aggregation Function: Average

Figure 12-45 shows the Order Fulfillment Time measure.

Cube	Measure	Source	Aggregation Function
OrderHandlingFuture1 Cube	Order Count	Order_Count Fact	Sum
	Order Price Total	Order Price Fact	Sum
	Order Price Average	Order Price Fact	Average
	Order Fulfillment Time	Order Process Time Fact	Average
OrderHandlingFuture1_InputCriterio...		OrderHandlingFuture1_InputCrit...	

Figure 12-45 Order Fulfillment Time measure

Create the KPI

KPIs are created in the KPI Model tab:

- ▶ Select ClipsAndTacksF1BMP and *New* → *KPI Context*. (Figure 12-46).
- ▶ Type ClipsAndTacks as the name of the context.
- ▶ Select ClipsAndTacks and *New* → *KPI*.
- ▶ Type Average Shipped Order Fulfillment Time as the name.
- ▶ Select *Duration* for the type (Figure 12-46).

KPI Details
Edit the details of the KPI, which is a performance measurement used to track business objectives.


ID: Edit...

Name:

Description:

Type: Duration

Figure 12-46 KPI Details

- ▶ Scroll down to KPI Target and Ranges.
- ▶ For Target, click *Details* and select 3 days.
- ▶ Select *Actual Value* for range.
- ▶ Click *Add* to add a range. Change the name to Range1 and click *OK*.
- ▶ Click  in the End value column and select 1 day.
- ▶ Add three more ranges with end times of 2, 3, and 4 days.
- ▶ Once all the ranges are added, the KPI Targets and Ranges section is shown in Figure 12-47.

▼ **KPI Target and Ranges**

Specify a target, which is an exact value for the KPI to achieve, or ranges against which to track the KPI, or both.

Target: Details...

Ranges: Details...

Range name	Start value	End value
Range1	0 Milliseconds	< 1 Days
Range2	1 Days	< 2 Days
Range3	2 Days	< 3 Days
Range4	3 Days	< 4 Days

Add Remove Sort

Figure 12-47 KPI Target and Ranges

- ▶ Scroll down to KPI Definition (Figure 12-48).
- ▶ For Cube, click *Browse* and select *OrderHandlingFuture1 Cube*.
- ▶ For Measures, click *Browse* and select *Order Fulfillment Time*.
- ▶ For Time dimension, click *Browse* and select *Start Time*.
- ▶ For Specify time period, select *Rolling*:
 - **Repeating**—To evaluate data for the full period or period in progress. For example, if you select daily, then the KPI can either be based on the values up to yesterday or be based on the values that indicate the day so far.
 - **Rolling**—Specifies whether to evaluate data from the period ending yesterday or the period up to present time. For example, if you select 30 days, the value of the KPI is based on the 30 days up to yesterday or up to the current time. The number of days cannot be more than 100 days for rolling dates.
 - **Fixed**—Specifies a time interval between start date and end date. Date ranges must fall between the beginning of 1995 and the end of 2009 (this can be edited by adding values in the Data Mart database).
- ▶ For Size of period, type 7 (days).
- ▶ For Base last day on, select *Period in progress*.
- ▶ For Dimensions, click *Add* and select *Shipped Flag* → *Shipped*.
- ▶ Expand *Shipped Flag* → *Shipped* and type SHIPPED as the value of the Shipped attribute.

Note: Do not use quotes around the value SHIPPED.

KPI Definition
Specify how the value of the KPI is set.

Cube:

Measure:

Time dimension:

Specify time period Repeating Rolling Fixed

Size of period days

Base last day on Last full period Period in progress

Dimensions:

Name	Attribute	Type	Value
Shipped ...	Shipped	String	SHIPPED

Note: This is a deviation from the definition of the KPI in the Modeler, where we did not define a rolling time period.

Figure 12-48 KPI Definition details

Percentage of orders shipped

The Shipped Order % KPI calculates the number of orders shipped versus the total number of orders.

Create Shipped Order % metric and measure

Create the Shipped Order % metric based on the Order_Shipped trigger as shown in Table 12-10.

Table 12-10 Shipped Order % metric details

Activity	Description / Location	Name / Value
Create metric	Metric Location: OrderHandlingFuture1	Name: Shipped Order % Type: Decimal Default Value: 0.00
Associate trigger to metric	Trigger Location: OrderHandlingFuture1	Metric Value Maps: Trigger: Order_Shipped Expression: integer(100) (select integer() under Functions)
Add measure	Data Mart Model → Measures: OrderHandlingFuture1 Cube	Name: Shipped Order % Source: Shipped Order % Fact Aggregation Function: Average

Figure 12-49 shows the metric (note the ID that is generated).

▼ Metric Details
 Edit the details of the metric, which is a holding spot for information used in other calculations.

ID:

Name:

Description:

Type:

Default Value:

This metric can be used for sorting

▼ Metric Value Maps
 Specify the expressions that set the value of the metric. If a trigger is specified, the map is evaluated when the trigger fires.

Trigger	Expression
Order_Shipped	integer(100)

How does this work?

- ▶ All orders get an initial value of 0
- ▶ Shipped orders get a value of 100
- ▶ The average is the percentage of shipped orders

Figure 12-49 shipped Order % metric details

Figure 12-50 shows the resulting measure.

▼ Measures
 Add measures, which are calculations performed on facts to collect and combine them into a single value.

Cube	Measure	Source	Aggregation Function
	Order Count	Order_Count Fact	Sum
	Order Price Total	Order Price Fact	Sum
	Order Price Average	Order Price Fact	Average
	Order Fulfillment Time	Order Process Time Fact	Average
	Shipped Order %	Shipped Order % Fact	Average
OrderHandlingFuture1_InputCriteria...		OrderHandlingFuture1_InputCrit...	

Figure 12-50 Measures for OrderHandlingFuture1 Cube

Create the KPI

Once the necessary metric and measure are created, we create the KPI.

- ▶ Create a KPI with the name Shipped Order % (Figure 12-51).
- ▶ Select *Decimal* as type.
- ▶ Specify 90 as target.
- ▶ Select *Actual value* for the range.
- ▶ Add four ranges with end values of 85, 90, 95, and 100.

▼ KPI Details
 Edit the details of the KPI, which is a performance measurement used to track business objectives.

ID:

Name:

Description:

Type:

▼ KPI Target and Ranges
 Specify a target, which is an exact value for the KPI to achieve, or ranges against which to track the KPI, or both.

Target:

Ranges:

Range name	Start value	End value
Range1	0	< 85
Range2	85	< 90
Range3	90	< 95
Range4	95	< 100

Figure 12-51 Shipped Order % KPI details

Specify the KPI Definition (Figure 12-52).

- ▶ Measure: Shipped Order %
- ▶ Time dimension: Start Time
- ▶ Specify time period: *Rolling*
- ▶ Size of period: 30
- ▶ Base period on: *Period in progress*

▼ KPI Definition
 Specify how the value of the KPI is set.

Cube:

Measure:

Time dimension:

Specify time period Repeating Rolling Fixed

Size of period Last days

Base last day on Last full period Period in progress

Dimensions:	Name	Attribute	Type	Value

Figure 12-52 Shipped Order % KPI Definition



Creating alerts

This section describes creating alerts to the respective business team when the stated business goals are not achieved.


Shipped order fulfillment time is greater than 3 days and 1 hour

To create an alert we have to create an event definition that contains the information to be sent to the business team. We send the order number of an order that exceeds 3 days and 1 hour.

In the Project Explorer select *ClipsAndTacksF1BMP* → *Event definitions* and *New* → *Event Definition*:

- ▶ Specify the name of the event as `OrderFulfillmentTime`.
- ▶ In the editor select *Property* click the *Add Property* icon .
- ▶ Specify the name as `OrderFulfillmentTime`.
- ▶ Similarly select *Extended Data* bar and click *Add Extended Data* , and specify the name as `Order` and leave the type as `noValue`.
- ▶ Create an other *Extended Data* item named `BusinessSituationName` and specify the type as `string`.

Note: The item `BusinessSituationName` is mandatory for the Monitor. The value specified for this extended data item is used by the Monitor to filter the event and take an appropriate action.

- ▶ Now select the `Order` item created and click *Add Extended Data As Nested Child* , and specify the name as `OrderNumber` and select `int` for the type. Add another item named `OrderDuration` with type `string` (Figure 12-53).

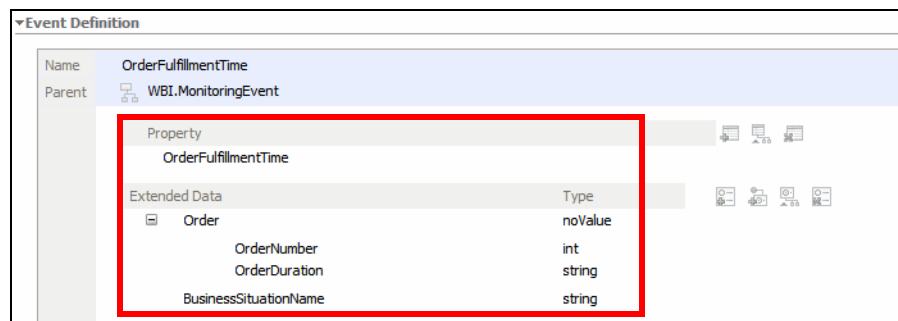


Figure 12-53 Event definition

This event can be generated for every order instance. We create the configuration steps in the Monitor Detail Model tab. Other alerts are defined on KPIs, therefore configuration steps will be done in the KPI Model tab.

Order Shipped Time metric

Create a metric for the order shipped time that captures the time when the order is shipped as shown in Table 12-11.

Table 12-11 Order Shipped Time metric details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: UpdateOrderDatabase_InputCriterion.	Name: Shipped_Time
Associate event to trigger	Trigger Source: UpdateOrderDatabase_InputCriterion.	Select: UpdateOrderDatabase_InputCriterion EXIT
Create metric	Metric Location: UpdateOrderDatabase_InputCriterion.	Name: Order Shipped Time Type: DateTime
Associate trigger to metric	Trigger Location: UpdateOrderDatabase_InputCriterion.	Metric Value Maps: Trigger: Shipped_Time Expression: UpdateOrderDatabase_InputCriterion_EXIT /predefinedData/creationTime

Fulfillment Time metric

Create a metric for the fulfillment time that calculates the time from the start of the process to the end of the process after shipment (Table 12-12).

Table 12-12 Fulfillment Time metric details

Activity	Description / Location	Name / Value
Create metric	Metric Location: UpdateOrderDatabase_InputCriterion	Name: Fulfillment Time Type: Duration Default Value: none
Associate trigger to metric	Trigger Location: UpdateOrderDatabase_InputCriterion	Metric Value Maps: Trigger: Shipped_Time Expression: Order Shipped Time - Start Time (Start Time is the Metric located at OrderHandlingFuture1)

Figure 12-54 shows the creation of Fulfillment Time metric. Notice that Start Time becomes ../Start_Time because it is at the parent level.

▼ Metric Details
 Edit the details of the metric, which is a holding spot for information used in other calculations.

ID:

Name:

Description:

Type:

Default Value:

This metric can be used for sorting

▼ Metric Value Maps
 Specify the expressions that set the value of the metric. If a trigger is specified, the map is evaluated when the trigger fires.

Trigger	Expression
Shipped_Time	Order_Shipped_Time - ../Start_Time

Figure 12-54 Fulfillment Time metric details

Order Number metric

Create a metric for the order number so that we can pass the number in the outbound event (Table 12-13).

Table 12-13 Order Number metric details

Activity	Description / Location	Name / Value
Create metric	Metric Location: OrderHandlingFuture1	Name: Order Number Type: Integer Default Value: none
Associate trigger to metric	Trigger Location: OrderHandlingFuture1	Metric Value Maps: Trigger: OrderVariable_Changed Expression: OrderVariable_CHANGED/extendedData/ variableData/OrderNumber

Elapsed_Duration_Time trigger

Create a trigger that is forced when the fulfillment time exceeds 3 days and 1 hour (Table 12-14).

Table 12-14 Elapsed_Duration_Time trigger details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: UpdateOrderDatabase_InputCriterion	Name: Elapsed_Duration_Time
Associate metric to trigger	Trigger Source: UpdateOrderDatabase_InputCriterion	Select: Fulfillment Time
Trigger condition	Trigger Condition: UpdateOrderDatabase_InputCriterion	Value: Fulfillment_Time > duration('P3DT1H0M0.000S') ^a

a. This value stands for 3 days (3D) and 1 hour (1H)

Figure 12-55 shows the creation of the Elapsed_Duration_Time trigger.

Trigger Details
Edit the details of the trigger, which detects an occurrence and initiates an action in response.

ID:

Name:

Description:

Trigger is repeatable
 Terminate monitoring context

Trigger Sources
Specify the source of this trigger.

Source Type	Source
Value change	Fulfillment Time

Trigger Condition
Specify the condition that determines whether the trigger will fire.

Figure 12-55 Trigger for elapsed time duration

Outbound event

We create an event that uses the event definition:

- ▶ Select *UpdateOrderDatabase_InputCriterion* and *New* → *Outbound Event*.
- ▶ Name the outbound event *Elapsed Duration Alert*.

- ▶ In the Outbound Event Details click *Browse* for Type. Select the *ClipsAndTacksF1BMP* → *OrderFulfillmentTime.cbe* → *OrderFulfillmentTime* event definition (Figure 12-56).

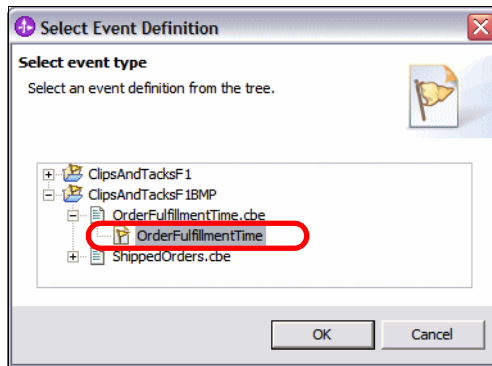


Figure 12-56 Select the event for the alert

- ▶ Click *Add* under Event Details Attributes and select the *Elapsed_Duration_Time* trigger.
- ▶ Expand the trigger sections (Figure 12-57).

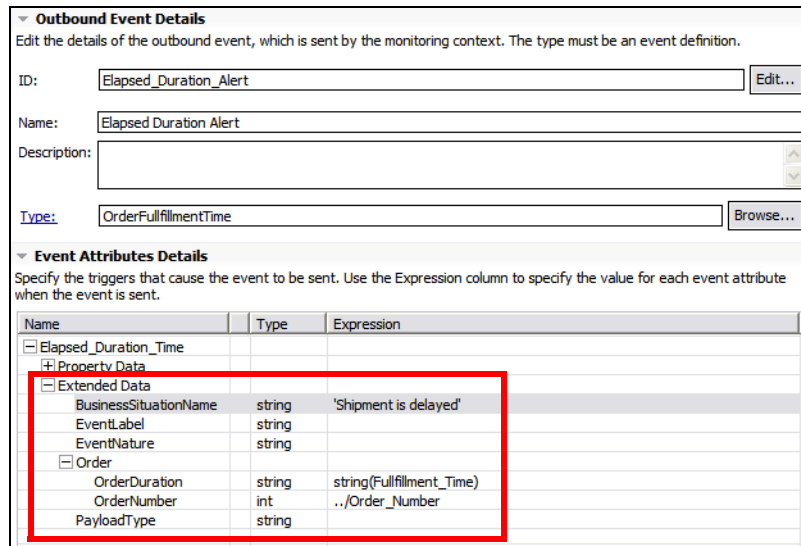


Figure 12-57 Outbound event with attributes

- ▶ Select *BusinessSituationName* and in the Expression field type 'Shipment is delayed'.

- ▶ Select *Order* → *OrderDuration* and in the Expression field select *ClipsAndTacksF1BMP* → *OrderHandlingFuture1* → *UpdateOrderDatabase_InputCriterion* → *Fulfillment Time*.

Enclose the Fulfillment Time with a `string()` function to match the target attribute type.

- ▶ Select *Order* → *OrderNumber* and in the Expression field select *ClipsAndTacksF1BMP* → *OrderHandlingFuture1* → *OrderNumber*.

Percentage of shipped orders are less than 85%

This section describes how to create an alert to the business team if the number of orders shipped falls to less than 85% of total orders.

We use the KPI that calculates the percentage of shipped orders and create the alert (Table 12-15).

Table 12-15 *Shipped Order less than 85% details*

Activity	Description / Location	Name / Value
Create event definition	Event Location: <i>Project Explorer:</i> <i>ClipsAndTacksF1BMP</i> → <i>New</i> → <i>Event Definition</i>	Name: ShippedOrders Property: ShippedOrders Extended Data: PercentageOfShippedOrders (float) BusinessSituationName (string)
Create trigger	KPI Model tab: <i>ClipsAndTacksF1BMP</i> → <i>ClipsAndTacks</i>	Name: ShippedOrder Trigger Sources: Click <i>Add</i> and Select <i>Recurring wait time</i> Trigger Condition: Shipped_Order__x0025_ < 85 (Select <i>ClipsAndTacksF1BMP</i> → <i>ClipsAndTacks</i> → <i>Shipped Order %</i> (result is Shipped_Order__x0025_) add < 85).

Activity	Description / Location	Name / Value
Create outbound event	KPI Model tab: ClipsAndTacksBMP → ClipsAndTacks	Name: ShippedOrdersLessThan85 Type: ShippedOrders (select ClipsAndTacksF1BMP → ShippedOrder.cbe → ShippedOrders) Event Attribute Details: Click Add and select ClipsAndTacks → ShippedOrders. Expression: expand Extended data PercentageOfShippedOrders: ClipsAndTacksF1BMP → ClipsAndTacks → Shipped Order %. BusinessSituationName: 'Shipped Orders percentage too low'

Figure 12-58 shows the event definition.

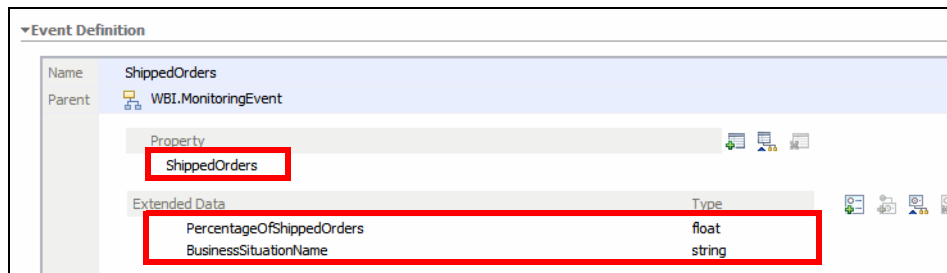


Figure 12-58 Event definition properties

Figure 12-59 shows the trigger for the alert.

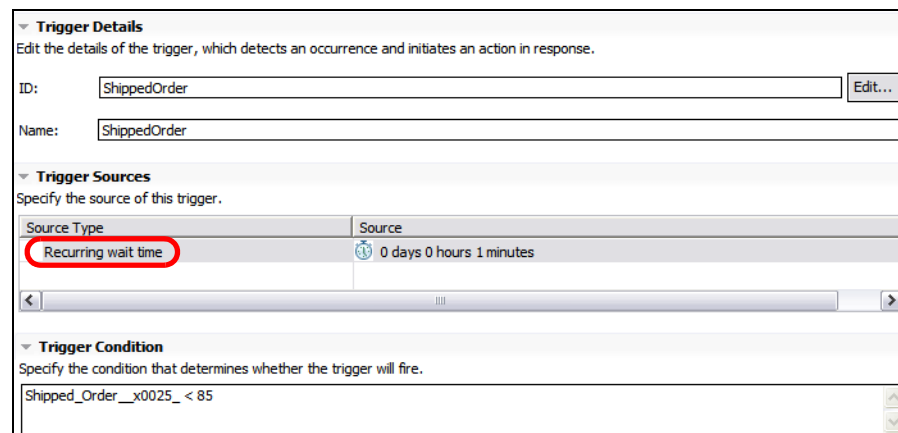


Figure 12-59 ShippedOrder trigger

The outbound event is shown in Figure 12-60.

Outbound Event Details
Edit the details of the outbound event, which is sent by the monitoring context. The type must be an event definition.

ID:

Name:

Description:

Type:

Event Attributes Details
Specify the triggers that cause the event to be sent. Use the Expression column to specify the value for each event attribute when the event is sent.

Name	Type	Expression
<input type="checkbox"/> ShippedOrder		
<input type="checkbox"/> Property Data		
ECSCurrentID	string	
ECSParentID	string	
ShippedOrders	string	
WBISSESSION_ID	string	
<input type="checkbox"/> Extended Data		
BusinessSituationName	string	'Shipped Orders percentage too low'
EventLabel	string	
EventNature	string	
PayloadType	string	
PercentageOfShippe...	float	Shipped_Order_x0025...

Figure 12-60 Shipped Order outbound event

Implementing round-trip measures

This section describes the configuration steps required in the Monitor Server to provide feedback to the Business Modeler.

We describe how to configure the Monitor model to retrieve the duration times of each activity and the decision percentages.

For all activity duration, a measure is created to calculate the time between the entry event and exit event of that particular activity. To calculate the decision activity percentages, a measure is created to get the actual percentage of each branch.

Determining activity duration

Let us start of with the Cancel Order process time. We create a measure to determine the time taken to cancel an order.

Create a stopwatch

Select *ClipsAndTacksF1BMP* → *OrderhandlingFuture1* → *CancelOrderandSendNotification_InputCriterion* and *New* → *Stopwatch*:

- ▶ Name the stopwatch *Cancel Order Processing Time*.
- ▶ For Stopwatch Controls, add two triggers using the *CancelOrder_Entry* trigger as the start, and the *Termination Trigger* as the stop (Figure 12-61).

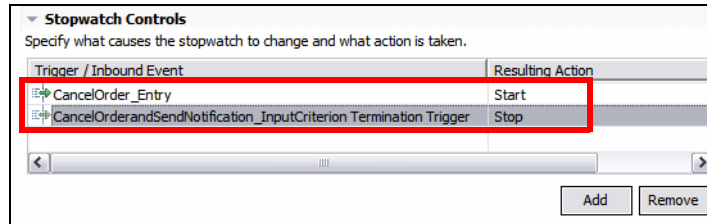


Figure 12-61 Stopwatch controls for activity duration

Create a measure

In the Data Mart Model add a measure to *CancelOrderandSendNotification*:

- ▶ Specify the name as *Cancel Order and Send Notification Processing Time*.
- ▶ Select the *Cancel Order Processing Time Fact* for the source for this measure and leave the function as *Average* (Figure 12-62).

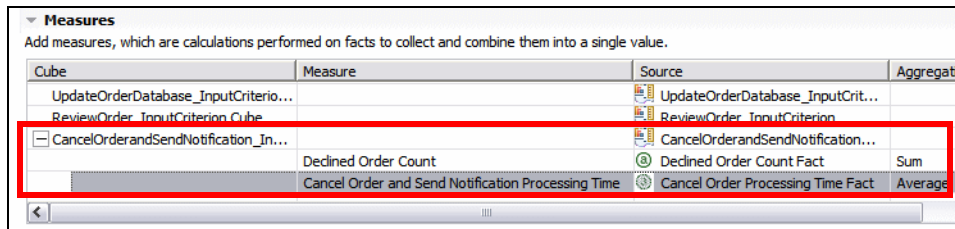


Figure 12-62 Measure for activity duration

Similarly, create stopwatches and measures for three other activities:

- ▶ *CheckCustomerAccountStatus_InputCriterion*
Check Customer Account Status Processing Time
- ▶ *CheckOrderHandlingPolicyforAutomaticApproval_InputCriterion*
Check Order Handling Policy Processing Time
- ▶ *UpdateOrderDatabase_InputCriterion*
Update Order Database Processing Time

Use the ENTRY event for the stopwatch start and the Termination Trigger for the stop.

For the human tasks, we cannot use this simple approach because the BPEL events of the process do not give us the information when a human task is claimed for processing. We have to implement separate monitor models for the two human tasks (see “Implementing round-trip measures for human tasks” on page 372).

Determining branch percentages

This section describes how to determine the percentage of orders that went through the Yes branch of the Approve Without Review decision.

If we determine the number of orders that went through the Check Customer Account Status activity out of all the orders that were processed, we can determine the percentage of orders that were approved without review.

We create a measure to count the number of orders that pass the Check Customer Activity Status activity:

- Specify Approve Without Review % metric details as specified in Table 12-16.

Table 12-16 Approve Without Review % details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: OrderHandlingFuture1	Name: CheckCustomerAccountStatus Entry
Associate event to trigger	Trigger Source: CheckCustomerAccountStatus_InputCriterion.	Select: CheckCustomerAccountStatus_InputCriterion ENTRY
Create metric	Metric Location: OrderHandlingFuture1	Name: Approve Without Review % Type: Decimal Default Value: 0.00
Associate trigger to metric	Trigger Location: CheckCustomerAccountStatus_InputCriterion.	Metric Value Maps: Trigger: CheckCustomerAccountStatus_Entry. Expression: integer(100)
Create measure	Date Mart Model: OrderHandlingFuture1 Cube	Name: Approve Without Review % Source: Approve Without Review % Fact Function: Average

We did not create the model definitions to measure the other branches.

Implementing round-trip measures for human tasks

In the ClipsAndTacks model, we use stand-alone human tasks (Review Order and Ship Order). To measure the task duration requires special steps to capture the events when a human task is claimed and finished.

The monitor model that we have created only consumes events generated by the Order Handling BPEL process, but not from the stand-alone human tasks that are invoked from the process.

To calculate the processing time of the human activity, we have to process the events that represent the time of claiming the task and the different situations that stop the processing of the human task.

There are different approaches to perform the processing time calculation:

- ▶ Use inline human tasks instead of a stand-alone human task. This would require major rework of the process that is generated from the Modeler.
- ▶ Generate events for the human task and integrate them to the monitor model that was generated for the process. Then create the event entries to consume the events and calculate the working duration.
- ▶ Generate separate monitor models for each human task. These separate models are configured to consume the events that represent the status of the human task. Then create the event entries to consume the events and calculate the working duration.

For ClipsAndTacks, we implement the third option by creating independent monitor models for the Review Order and Ship Order human tasks. However, instead of deploying these models separately, we copy the relevant portions of the generated model—the monitoring context and the events—into the main model and then develop the business measures.

Creating the monitor models for human tasks

We have to generate a separate business measures model for the Review Order and Ship Order to Customer human tasks.

Create the human tasks events for Review and Ship Order

In the Business Integration view, open the Review Order human task in the editor. In the Properties view, Event Monitor tab, select *All* and *Full* (Figure 12-63). Note that these human task events are different from the BPEL events that we specified earlier in Figure 12-10 on page 335.

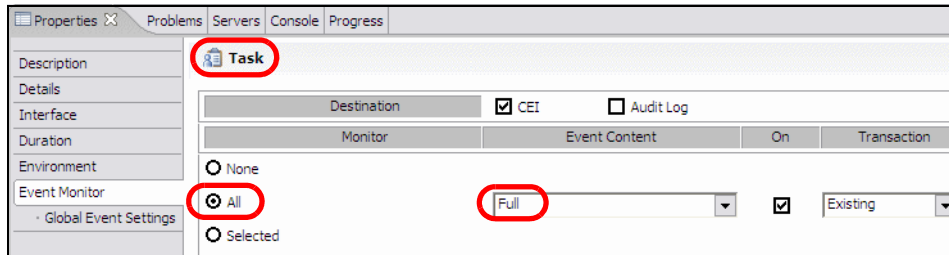


Figure 12-63 Specify events for human tasks

Do the same for the Ship Order to Customer human task.

Generate the human task events

In the Business Integration view select the Review Order human task and *Monitor Tools* → *Generate Event Definitions*.

This step creates additional events that are visible in the Business Monitoring perspective (for example, BPC.HTM.BASE, BPC.HTM.TASK.BASE, and so forth).

Then do the same for Ship Order to Customer and one new event is added.

Generate the monitor models

In the Business Integration view select the Review Order human task and *Monitor Tools* → *Generate Monitor Model*.

In the dialog select the same monitor project (C1 ipsAndTacksF1BMP), but use a different monitor model name, for example, **C1 ipsAndTacksF1ReviewBMP**. Do not use the name of the existing model because it would be destroyed.

Generate the model for Ship Order to Customer in the same way and name it **C1 ipsAndTacksF1ShipBMP**.

Merging the monitor models

Merging the models is a manual task. However, it is worth the effort to spend time now to merge the models into a single model, rather than spending more time later to deploy multiple models.

Monitor model layout of XML file

Let us understand the layout of a model XML file. Open the C1 ipsAndTacksF1BMP monitor model at the .mm tab and study the sections (Example 12-1).

Example 12-1 Layout of a monitor model

```
<monitor>
  <monitorDetailsModel>
    <monitoringContext>
      ... embedded contexts of activities
    </monitoringContext>
  </monitorDetailsModel>
  <kpiModel>
    ... kpiContext, trigger, outbound event, kpi
  </kpiModel>
  <dataMartModel>
    <cube>
      ... facts, measures, dimensions
    </cube>
    <cube.....>
  </dataMartModel>
  <visualModel>
    ... not present in generated model, but is there in model from Modeler
  </visualModel>
  <eventModel>
    <import .....>
    <import .....>
  </eventModel>
</monitor>
```

To merge the human task models into the main model, we have to copy the sections into the appropriate place in the main model (Example 12-2).

Example 12-2 Layout of the monitor model after copying from other models

```
<monitor>
  <monitorDetailsModel>
    <monitoringContext>
      ... embedded contexts of activities
    </monitoringContext>
    <monitoringContext>
      ... copied from ClipsAndTacksF1ReviewBMP
    </monitoringContext>
    <monitoringContext>
      ... copied from ClipsAndTacksF1ShipBMP
    </monitoringContext>
  </monitorDetailsModel>
  <kpiModel>
    ... kpiContext, trigger, outbound event, kpi
  </kpiModel>
  <dataMartModel>
    <cube>
      ... facts, measures, dimensions
```



```

    </cube>
    <cube.....>
    <cube>
        ... copied from ClipsAndTacksF1ReviewBMP
    </cube>
    <cube>
        ... copied from ClipsAndTacksF1ShipBMP
    </cube>
</dataMartModel>
<visualModel>
    ... copied from ClipsAndTacksF1mm (imported from Modeler)
</visualModel>
<eventModel>
    <import .....>
    <import .....>
    <import ..... copied from ClipsAndTacksF1ReviewBMP>
    <import .....>
    <import .....copied from ClipsAndTacksF1ShipBMP>
</eventModel>
</monitor>

```

Before merging the models, make a copy of the ClipsAndTacksF1BMP model (select the model and *Copy*, then select *Monitor Models* and *Paste*).

Copy the Review Order model into the main model

Open both the ClipsAndTacksF1BMP and ClipsAndTacksF1ReviewBMP models and go to the .mm tab:

- ▶ In the review model, select the complete <monitoringContext> section and *Copy*.
- ▶ In the main model, place the cursor just before the end of the details section (</monitorDetailsModel>) and *Paste*.
- ▶ In the review model, select the complete <cube> section and *Copy*.
- ▶ In the main model, place the cursor just before the end of the data mart section (</dataMartModel>) and *Paste*.
- ▶ In the review model, select all the <import> lines and *Copy*.
- ▶ In the main model, place the cursor just before the end of the event section (</eventModel>) and *Paste*.

Study all the event import lines. One of the copied lines is a duplicate:

```
<import location="/ClipsAndTacksF1/events/WBI.MonitoringEvent.cbe"/>
```

This event appears in every model and should be there only once. Therefore, delete the second occurrence of the WBI.MonitoringEvent.cbe event.

Copy the Ship Order model into the main model

Open both the ClipsAndTacksF1BMP and ClipsAndTacksF1ShipBMP models and go to the .mm tab:

- ▶ Copy the <monitoringContext> section to the main model.
- ▶ Copy the <cube> section to the main model.
- ▶ Study the event <import> lines in the ClipsAndTacksF1ShipBMP model. The list of events is the same as in the Review Order model, except for one event:

```
<import location="/ClipsAndTacksF1/events/BPC.HTM.TASK.MESSAGE_
ClipsAndTacksF1_processes_orderhandlingfuture1_
ShipOrdertoCustomer_0567294367_t_
ShipOrdertoCustomer_0567294367.cbe"/>
```

- ▶ Select only the one event that is different from the main model.

Save the merged model. There should be no errors.

Copy the visual model into the main model

The visual model is available in the ClipsAndTacksF1mm project that we imported from the Modeler.

Open both the ClipsAndTacksF1BMP and the Order Handling (Future 1) Business Measures (in ClipsAndTacksF1mm) models and go to the .mm tab:

- ▶ In the imported model, select the complete <visualModel> section and *Copy*.
- ▶ In the main model, place the cursor between the data mart and the event sections (</dataMartModel>**X**<eventModel>) and *Paste*.
- ▶ Save the merged model. There are two problems in the <visualization> tags:

```
<visualization
  context="/Order_Handling__x0028_Future_1_x0029__Business_Measures
  /MDM/Order_Handling__x0028_Future_1_x0029__MC">
<visualization
  context="/Order_Handling__x0028_Future_1_x0029__Business_Measures
  /KM/Order_Handling__x0028_Future_1_x0029__KC">
```

- ▶ The first context points to the IDs of the monitor model, details model, and monitor context:

```
<monitor ..... id="OrderHandling...."
  <monitorDetailsModel displayName="..." id="MDM">
  <monitoringContext displayName="..." id="OrderHandling....">
```

- ▶ The second context points to the IDs of the monitor model, KPI model, and KPI context:

```
<monitor ..... id="OrderHandling....."
  <kpiModel displayName="...." id="KM">
    <kpiContext displayName="...." id="OrderHandling.....">
```

- ▶ We have to change the two lines with the IDs of our model:

```
<visualization context="/ClipsAndTacksF1BMP/MDM/OrderHandlingFuture1">
<visualization context="/ClipsAndTacksF1BMP/KM/ClipsAndTacks">
```

- ▶ Save the ClipsAndTacksF1BMP model. You should have no errors.

Verify the merged model

In the Monitor Details Model tab, you have three main contexts (Figure 12-64).

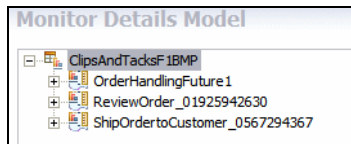


Figure 12-64 Monitor details of merged model

In the Data Mart Model tab, you can find two new cubes, and in the Visual Model tab, you can see the diagrams for the process and the KPIs.

Business measures for human task processing time

Create a stopwatch and a measure using the information in Table 12-17. The final measure is shown in Figure 12-65.

Table 12-17 Business measures for a human task

Activity	Description / Location	Name / Value
Create start trigger	Trigger Location: ReviewOrder_ xxxx	Name: StartStopwatch
Associate event to trigger	Trigger Source: ReviewOrder_ xxxx	Select: ReviewOrder_ xxxx ASSIGNED
Create stop trigger	Trigger Location: ReviewOrder_ xxxx	Name: StopStopWatch

Activity	Description / Location	Name / Value
Associate event to trigger Add 5 events	Trigger Source: ReviewOrder_xxxx	Select: ReviewOrder_xxxx DEASSIGNED ReviewOrder_xxx EXPIRED ReviewOrder_xxxx FAILED ReviewOrder_xxxx TERMINATED ReviewOrder_xxxx EXIT
Create stopwatch	Stopwatch Location: ReviewOrder_xxxx	Name: Review Order Duration Accumulating stopwatch: true Stopwatch Control: StartStopWatch → start StopStopWatch → stop
Add measure	Data Mart Model → Measures: ReveiwOrder_xxxx Cube	Name: Review Order Processing Time Source: Review Order Duration Fact Aggregation Function: Average

Measures			
Add measures, which are calculations performed on facts to collect and combine them into a single value.			
Cube	Measure	Source	Aggregation Function
[-] ReviewOrder_01925942630 Cube	Review Order Processing Time	[+] ReviewOrder_01925942630 [+] Review Order Duration Fact	Average

Figure 12-65 Review Order Processing Time measure

Steps for measuring the Ship Order processing time

Define the same triggers, stopwatch, and measure steps for the Ship Order To Customer human task:

- ▶ Create two triggers (StartStopwatch, StopStopwatch with five events)
- ▶ Create a stopwatch (Ship Order Duration, accumulating)
- ▶ Create a measure (Ship Order Processing Time).

The final measure is shown in Figure 12-66

Measures			
Add measures, which are calculations performed on facts to collect and combine them into a single value.			
Cube	Measure	Source	Aggregation Function
[-] ShipOrdertoCustomer_0567294367 ...	Ship Order Processing Time	[+] ShipOrdertoCustomer_0567294367 [+] Ship Order Duration Fact	Average

Figure 12-66 Ship Order Processing Time measure

Displaying business metrics and KPIs in the visual model

The visual model can be used to display instance metrics and KPIs in the graphical diagram. We did not make use of this functionality in the Future 1 implementation.

Refer to “Using the visual model” on page 564 for examples of using the visual model for graphical display of process instances and KPIs in the Monitor Dashboard for the Future 2 implementation.

Modeler feedback loop

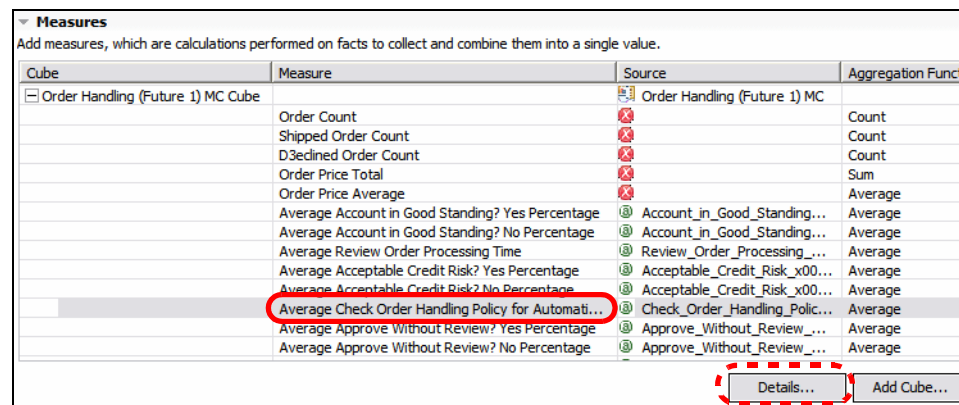
To feed the measurements of the task processing times and the decision percentages back to the Modeler, we have to identify the measures with a tracking key that the Modeler will understand to update the business model.

The imported model from the Modeler does contain the tracking keys in its measures. We have to copy these keys to the corresponding measure in our model.

Copying tracking keys for Modeler feedback

Expand the `ClipboardTasksFuture1` project in the Project Explorer and open the Order Handling (Future 1) Business Measures:

- ▶ Select the *Data Mart Model* tab, scroll-down to Measures, expand the *Order Handling (Future 1) MC Cube*, and select one of the activity duration measures: *Average Check Order Handling Policy for Automatic Approval Processing Time* (Figure 12-67).



Cube	Measure	Source	Aggregation Function
Order Handling (Future 1) MC Cube	Order Count	Order Handling (Future 1) MC	Count
	Shipped Order Count		Count
	Declined Order Count		Count
	Order Price Total		Sum
	Order Price Average		Average
	Average Account in Good Standing? Yes Percentage	Account_in_Good_Standi...	Average
	Average Account in Good Standing? No Percentage	Account_in_Good_Standi...	Average
	Average Review Order Processing Time	Review_Order_Processing_...	Average
	Average Acceptable Credit Risk? Yes Percentage	Acceptable_Credit_Risk_x00...	Average
	Average Acceptable Credit Risk? No Percentage	Acceptable_Credit_Risk_x00...	Average
	Average Check Order Handling Policy for Automatic Approval Processing Time	Check_Order_Handling_Polic...	Average
	Average Approve Without Review? Yes Percentage	Approve_Without_Review_...	Average
	Average Approve Without Review? No Percentage	Approve_Without_Review_...	Average

Figure 12-67 Round trip measure for an activity duration

- ▶ Click *Details*. Notice the tracking key (Figure 12-68).

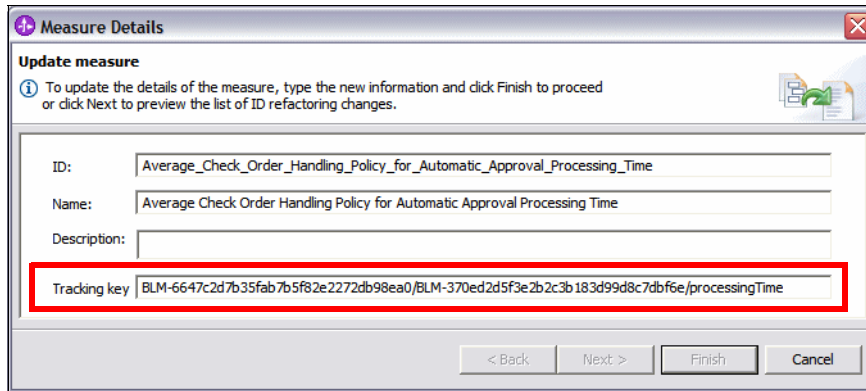


Figure 12-68 Round trip measure details with tracking key

- ▶ Select the tracking key and *Copy*, then click *Cancel*.
- ▶ Switch to the *ClipsAndTacksF1BMP* project. We have to find the corresponding measure in our model.
- ▶ Select the *Data Mart Model* tab, scroll down to *Measures* and select the *CheckOrderHandlingPolicyforAutomaticApproval_InputCriterion Cube* → *Check Order Handling Policy Processing Time* measure (Figure 12-69).

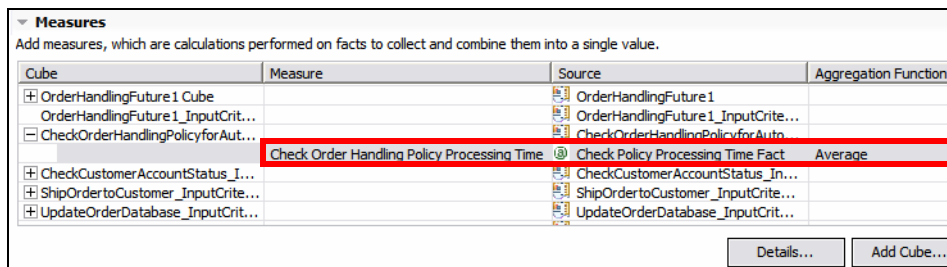


Figure 12-69 Check Order Handling Policy Processing Time measure

- ▶ Click *Details* and paste the tracking key into the measure. Click *Finish* to save the tracking key (Figure 12-70).

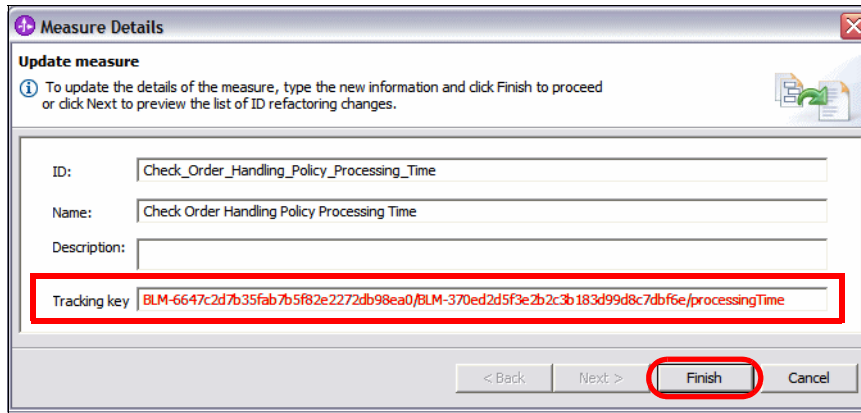


Figure 12-70 Copying the tracking key from another model

- ▶ With the measure still selected, switch to the *ClipsAndTacksF1BMP.mm* tab. In the XML file, you can find the measure highlighted. Scroll to the right and you can see the tracking key (Figure 12-71).

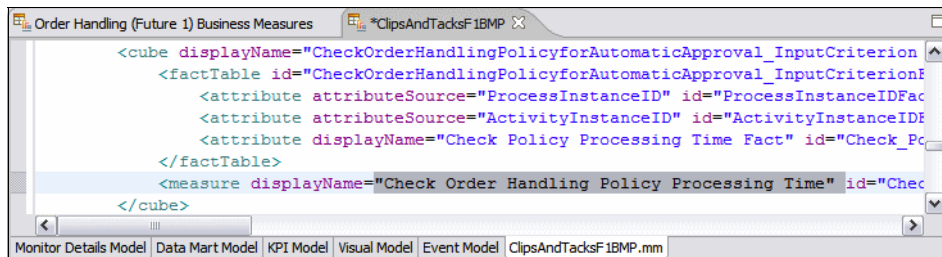


Figure 12-71 XML file with the measure and tracking key

Here is the coding:

```
<measure displayName="Check Order Handling Policy Processing Time"
  id="Check_Order_Handling_Policy_Processing_Time"
  aggregationType="avg"
  source="CheckOrderHandlingPolicy../Check_Policy_Processing_Time_Fact"
  trackingKey="BLM-6647c2d7b35fab7b5f82e2272db98ea0
  /BLM-370ed2d5f3e2b2c3b183d99d8c7dbf6e/processingTime">
```

What measures require a tracking key?

We did not define all the round trip measures in our model. So it might be easier to locate our measures and then find the tracking key in the model imported from the Modeler.

Table 12-18 lists our measures and the matching measure in the imported model where we can find the tracking key.

Table 12-18 Measures that require a tracking key

ClicksAndTacksF1BMP Cube → Measure	Order Handling () Business Measures → Measure (only one cube)
OrderHandlingFuture1 Cube ▶ Order Fulfillment Time ▶ Approve Without Review %	▶ Average Order Handling (Future 1) Processing Time ▶ Average Approve Without Review? Yes Percentage
CheckOrderHandlingPolicy... Cube ▶ Check Order Handling Policy Processing Time	▶ Average Check Order Handling Policy for Automatic Approval Processing Time
CheckCustomerAccountStatus... Cube ▶ Check Customer Account Status Processing Time	▶ Average Check Customer Account Status Processing Time
UpdateOrderDatabase... Cube ▶ Update Order Database Processing Time	▶ Average Update Order Database Processing Time
CancelOrderandSendNotification... Cube ▶ Cancel Order and Send Notification Processing Time	▶ Average Cancel Order and Send Notification Processing Time
ReviewOrder_xxxx Cube ▶ Review Order Processing Time	▶ Average Review Order Processing Time
ShipOrdertoCustomer_xxxx Cube ▶ Ship Order Processing Time	▶ Average Ship Order to Customer Processing Time

Monitor model content

Let us review the triggers, metrics, and measures that we defined (Table 12-19).

Table 12-19 Recapitulation of model elements

Component	Triggers	Metrics	Measures
OrderHandling Future1	<ul style="list-style-type: none"> ▶ CheckCustomer AccountStatus Entry ▶ Order_Entry ▶ Order_Exit ▶ Order_Shipped ▶ OrderVariable_Changed 	<ul style="list-style-type: none"> ▶ Approve Without Review % ▶ Order City ▶ Order Country ▶ Order Number ▶ Order Price ▶ Shipped Flag ▶ Shipped Order % ▶ Start Time ▶ Order_Count ▶ Order Process Time 	<ul style="list-style-type: none"> ▶ Order Count ▶ Order Price Total ▶ Order Price Average ▶ Order Fulfillment Time ▶ Shipped Order % ▶ Approve Without Review % <p>Dimensions:</p> <ul style="list-style-type: none"> ▶ Location ▶ Shipped Flag ▶ Start Time
CancelOrder...	<ul style="list-style-type: none"> ▶ CancelOrder_Entry 	<ul style="list-style-type: none"> ▶ Declined Order Count ▶ Cancel Order Processing Time 	<ul style="list-style-type: none"> ▶ Declined Order Count ▶ Cancel Order and Send Notification Processing Time
CheckCustom erAccount...		<ul style="list-style-type: none"> ▶ Check Customer Processing Time 	<ul style="list-style-type: none"> ▶ Check Customer Account Status Processing Time
Check..Policy...		<ul style="list-style-type: none"> ▶ Check Policy Processing Time 	<ul style="list-style-type: none"> ▶ Check Order Handling Policy Processing Time
ReviewOrder...			
ShipOrder...	<ul style="list-style-type: none"> ▶ ShipOrder_Exit 	<ul style="list-style-type: none"> ▶ Shipped Order Count 	<ul style="list-style-type: none"> ▶ Shipped Order Count
UpdateOrder...	<ul style="list-style-type: none"> ▶ Elapsed_Duration_Time ▶ Shipped_Time 	<ul style="list-style-type: none"> ▶ Fulfillment Time ▶ Order Shipped Time ▶ Update Order Database Processing Time ▶ Elapsed Duration Alert 	<ul style="list-style-type: none"> ▶ Update Order Database Processing Time
ReviewOrder_ xxxxxxxx.	<ul style="list-style-type: none"> ▶ StartStopWatch ▶ StopStopwatch 	<ul style="list-style-type: none"> ▶ Review Order Duration 	<ul style="list-style-type: none"> ▶ Review Order Processing Time
ShipOrdertoCu stomer_xxxxx	<ul style="list-style-type: none"> ▶ StartStopWatch ▶ StopStopwatch 	<ul style="list-style-type: none"> ▶ Ship Order Duration 	<ul style="list-style-type: none"> ▶ Ship Order Processing Time

In addition, we have defined two KPIs: Average Shipped Order Fulfillment Time (3 day target) and Shipped Order % (90% target).

Deploying the monitor model to the unit test environment

In this section we describe how to generate an EAR file from the monitor model, and the deployment process to deploy the EAR file to the Monitor unit test environment:

- ▶ Select *ClipsAndTacksF1BMP* → *Monitor Models* → *ClipsAndTacksF1BMP* and *Generate Monitor EAR*.
- ▶ Accept the names for the EJB and EAR projects (Figure 12-72).

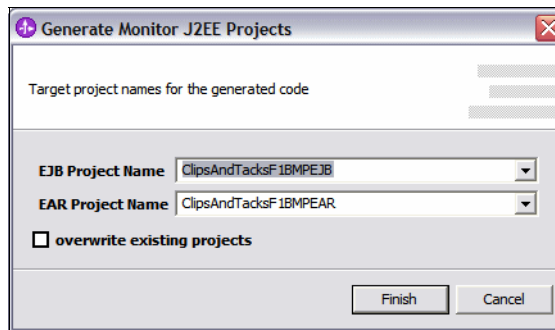


Figure 12-72 Name the EAR file

- ▶ Change to the Web perspective. The two generated projects are found under Enterprise Applications and EJB projects respectively.
- ▶ Select *EJB Projects* → *ClipsAndTacksF1BMPEJB* and *Deploy*.

Configuring the Monitor Server

In the Servers view, start the WebSphere Business Monitor Server v6.0.2. Make sure the Process Server is stopped (Figure 12-73).

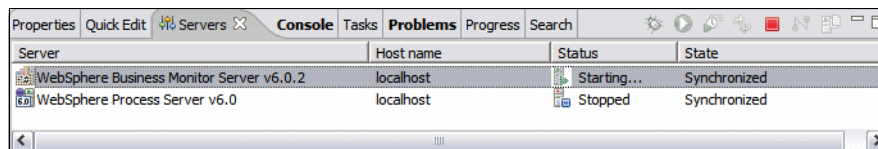


Figure 12-73 Start the Monitor Server unit test environment

Configuring security

If you run the application with security turned on, then you must configure the Monitor Server with the same security settings that were used to run the application in Process Server.

Note that the user ID used to install the Monitor Toolkit (see “Installation of the Monitor Development Toolkit”) must be defined in the user registry that is used. We installed the Monitor Toolkit with a user ID of **db2admin**, therefore, we added this ID to the `users.props` and `groups.props` files in the custom user registry.

The same user ID and password must be used when defining security and when defining how Integration Developer communicates with the server.

We had to configure one extra setting:

- ▶ In the administrative console, select *Global Security*. Expand *Authentication* → *JAAS Configuration* → *J2C Authentication data*.
- ▶ The user IDs for `MonitorAp1phab1oxA1ias` and `MonitorQueueConnectionFactoryAuth` were set to UNSET. We updated these two entries to use the same user ID and password as the other entries.

Deploying the business process application

The business process application consists of three enterprise applications:

- ▶ `ClipsAndTacksF1App`
- ▶ `ClipsAndTacksFxFHumanCustomEAR`
- ▶ `CrediServiceEAR`

Select the Monitor Server and *Add and remove projects*. Add the three enterprise applications to the server and wait until they are published.

Deploying the monitor model application

Select the Monitor Server and *Add and remove projects*. Add the `ClipsAndTacksF1BMP` project to the server (Figure 12-74).

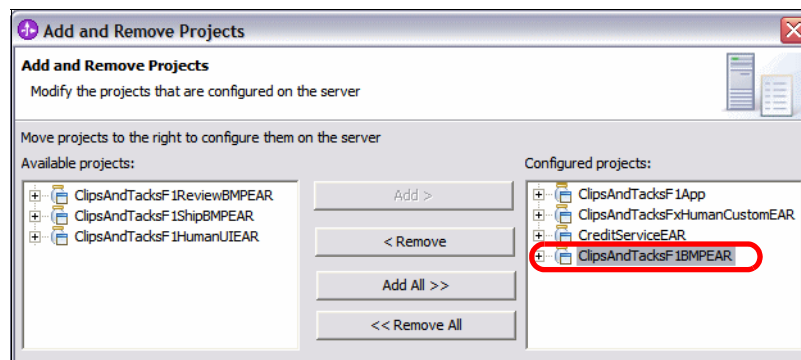


Figure 12-74 Deploy the Monitor EAR file

Notice that publishing fails. We have to do more configuration of the monitor model application.

Make the monitor model application startable

The monitor model application ClipsAndTacksF1BMP is not in a startable state:

- ▶ Select the Monitor Server and *Run administrative console*.
- ▶ Log on to the administrative console, navigate to *Applications* → *Monitor Models*. The ClipsAndTacksF1BMP project is deployed and the status is *not startable*. To change the measurement model to a *startable* state, a few configuration steps are required.
- ▶ Select the hyper link (2007-xx-xxTxx:xx:00) in the Version column to configure the model (Figure 12-75).

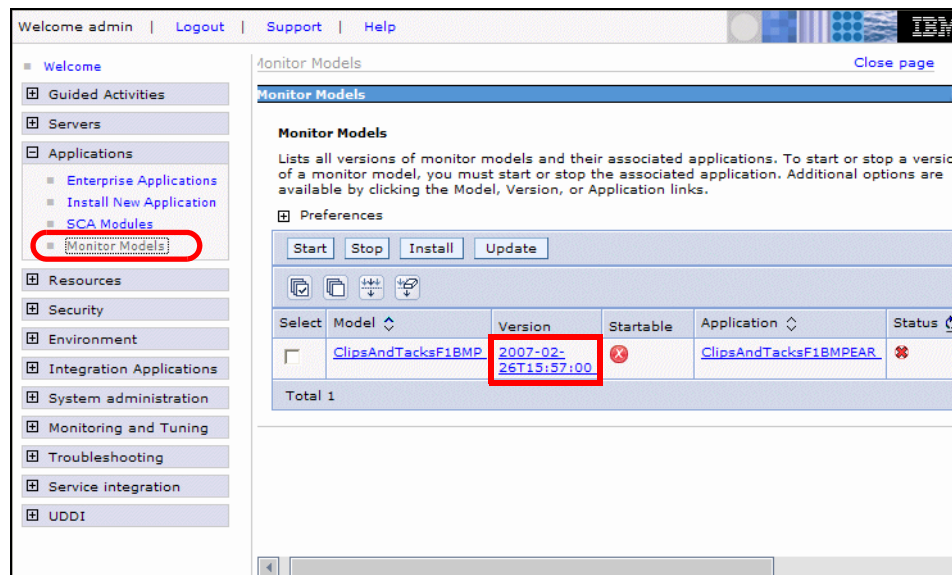


Figure 12-75 Monitor Models with project not startable

- ▶ Click *Setup Wizard* under Version Properties to continue the configuration (Figure 12-76).

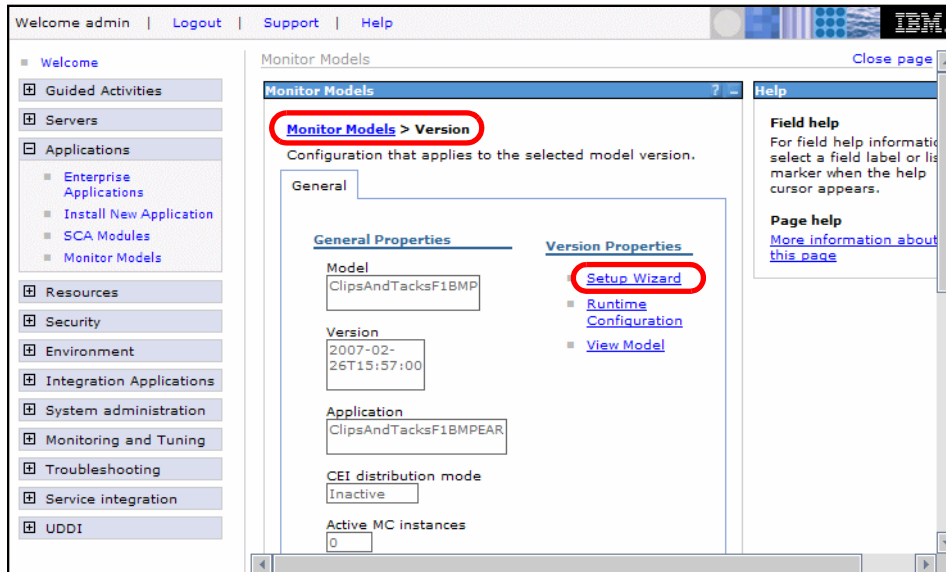


Figure 12-76 Setup Wizard for monitor model application

Step 1: Run Data Services Generation

Specify the output directory (Figure 12-77), for example:

C:\SG247148\sampcode\BMPScripts

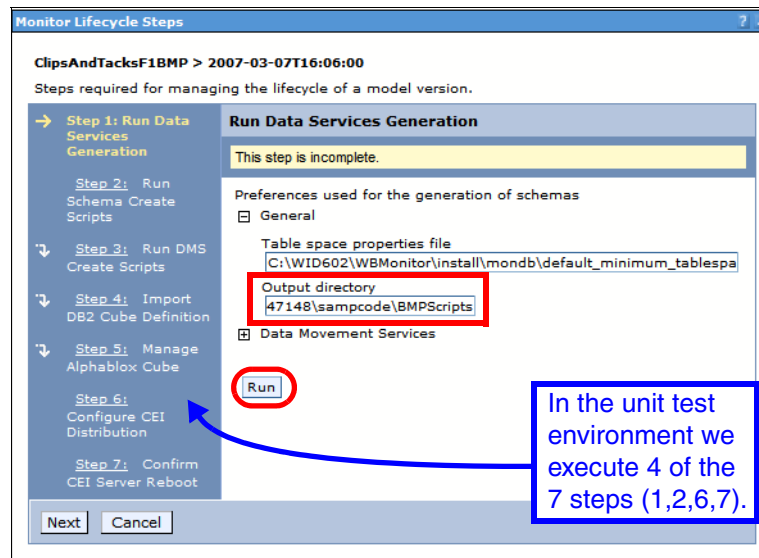


Figure 12-77 Step 1: Run Data Service Generation

Click *Run* to generate the scripts. You get a confirmation that the step is complete. Click *Next*.

Note: Explore the generated files: `state.ddl` creates the tables, `dropState.d11` removes the tables, and `stateMapping.html` describes the tables and their columns.

Step 2: Run Schema Create Scripts

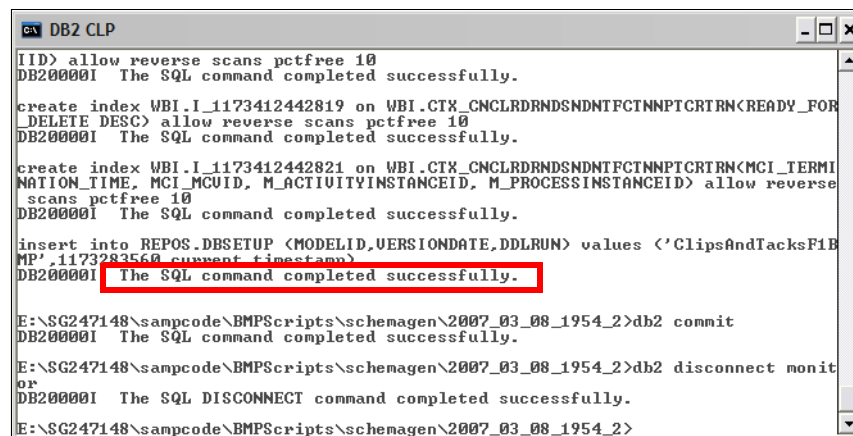
Read the directions and run the scripts in a DB2 command window.

- ▶ Open a DB2 command window.
- ▶ Navigate to the location where `state.ddl` file is generated:

```
C:\SG247148\sampcode\BMPScripts\schemagen\2007xxxxxx
```

- ▶ Run the commands (Figure 12-78):

```
db2 connect to monitor
db2 +c -stvf state.ddl
db2 commit
db2 disconnect monitor
```



```
DB2 CLP
[11D] allow reverse scans pctfree 10
DB20000I The SQL command completed successfully.

create index WBI.I_1173412442819 on WBI.CTX_CNCLDRDRNDSNDNTFCTNNPTCRTRM<READY_FOR_DELETE DESC> allow reverse scans pctfree 10
DB20000I The SQL command completed successfully.

create index WBI.I_1173412442821 on WBI.CTX_CNCLDRDRNDSNDNTFCTNNPTCRTRM<MCI_TERMINATION_TIME, MCI_MCUID, M_ACTIVITYINSTANCEID, M_PROCESSINSTANCEID> allow reverse scans pctfree 10
DB20000I The SQL command completed successfully.

insert into REPOS.DBSETUP (MODELID,VERSIONDATE,DDLRUN) values ('ClipsAndTacksFlBMP',1173283560,current_timestamp)
DB20000I The SQL command completed successfully.

E:\SG247148\sampcode\BMPScripts\schemagen\2007_03_08_1954_2>db2 commit
DB20000I The SQL command completed successfully.

E:\SG247148\sampcode\BMPScripts\schemagen\2007_03_08_1954_2>db2 disconnect monitor
DB20000I The SQL DISCONNECT command completed successfully.

E:\SG247148\sampcode\BMPScripts\schemagen\2007_03_08_1954_2>
```

Figure 12-78 Run the database DDL scripts

- ▶ In the administrative console click *Refresh* in Step 2 and the step is marked as complete.
- ▶ Click *Next* (Figure 12-79).

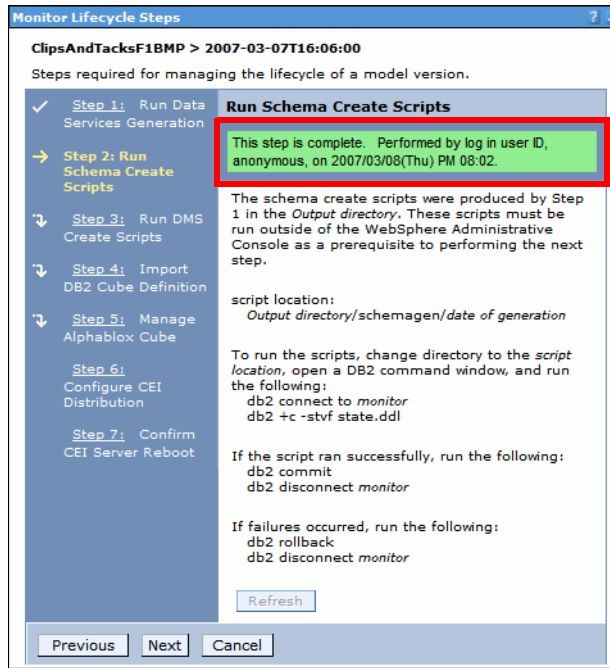


Figure 12-79 Step 2: Run Schema Create Scripts completed

Steps 3 to 5 are disabled in the test environment

Click **Next** until you reach step 6.

Step 6: Configure CEI Distribution

Click **Apply**, wait for the confirmation messages, and click **Next** (Figure 12-80).

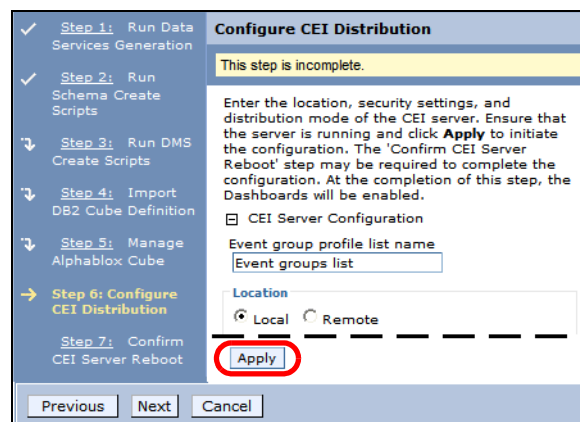
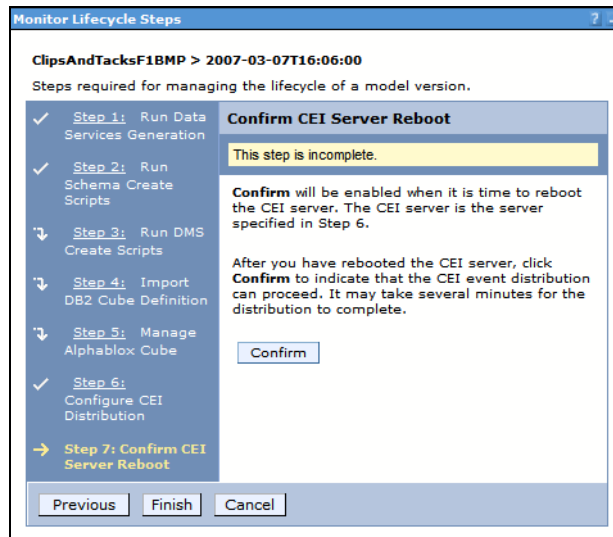


Figure 12-80 Step 6: Configure CEI Distribution

Step 7: Confirm CEI Server Reboot

After configuration of the CEI, we have to restart the CEI server (Figure 12-81). The CEI server is the server where the business process application runs, which in our case is the Monitor Server itself.



This step is eliminated with the installation of interim fix 1 or upgrade to 6.0.2.1 for the Monitor Toolkit

Figure 12-81 Step 7: Confirm CEI Server Reboot

- ▶ Click *Logout* and save the configuration when prompted.
- ▶ Select the WebSphere Monitor Server in the Servers view and *Restart* → *Start*.
- ▶ When the server is restarted, open the administrative console and login.
- ▶ If you did not save the configuration, you are prompted to recover the changes done in the previous session, click *OK* (Figure 12-82).

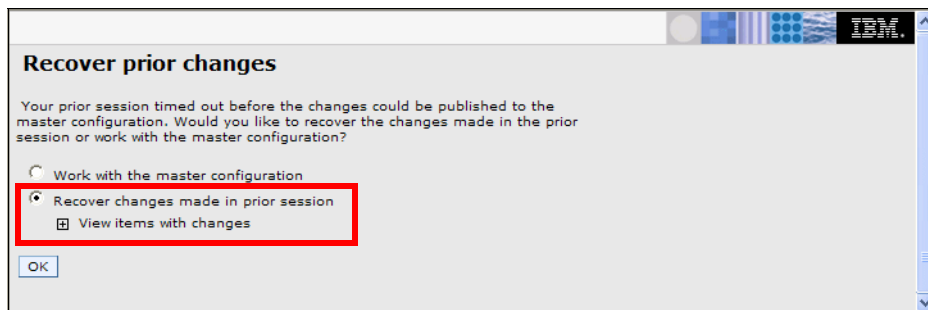


Figure 12-82 Recover changes made in prior session

- ▶ Navigate to *Applications* → *Monitor Models* → *2007-xxxxx* → *Setup Wizard* → *Step 7: Configure CEI Server Reboot*.
- ▶ Click *Confirm*, and Click *Finish*.

The model is now in a startable state and the services for the model are started (Figure 12-83). The ClipsAndTacksF1BMPEAR application is now started.

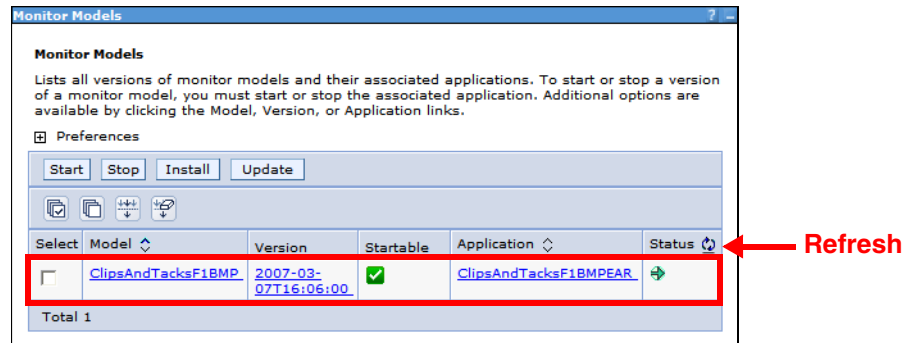



Figure 12-83 Monitor model after successful deployment

Note: This panel may not refresh quickly. Click *Refresh* ( icon) to update the status.

We have completed the configuration of the monitor model in the Monitor unit test environment.

Logout from the administrative console.

Defining the data source in the Monitor Server

The ClipsAndTacks application requires the data source for the CLIPTACK database. You can follow the instructions in “Creating a data source for the database” on page 218 or, preferred and faster, “Using a JACL script to define the data source” on page 404:

- ▶ Run the **dbadmin** command in <WID-HOME>\pf\wbmonitor\bin.
- ▶ Use the same database location as for testing: <WID-HOME>/pf/wps.

Testing the monitor model

To run the business process application, you have to make sure that you run it on the Monitor Server:

- ▶ Select the `ClipsAndTacksF1Invoke` project and *Properties*. Select *Server* and select the *WebSphere Business Monitor Server v6.0.2* and click *Apply* (Figure 12-84). Alternatively select `<None>` and you are prompted for the server every time you run the application.

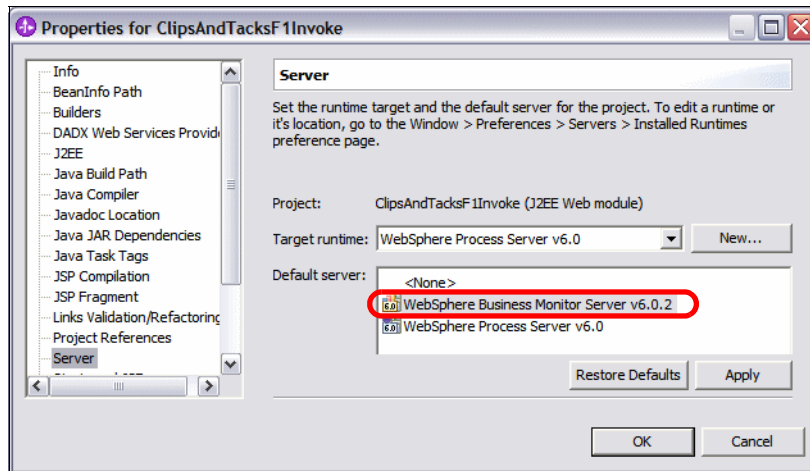


Figure 12-84 Project properties to run on the Monitor Server

- ▶ Do the same for the `ClipsAndTacksFxHumanCustomWeb` project.

Start the front-end and the human task application:

- ▶ Select the `ClipsAndTacksF1Invoke` project and *Run* → *Run on Server*.
- ▶ Select the `ClipsAndTacksFxHumanCustomWeb` project and *Run* → *Run on Server*. Notice that the port is different than the Process Server (9081).

To test the model that is generated, let us submit three orders.

1. Order with value less than \$750.
2. Order with value more than \$750 and reject the order.
3. Order with value more than \$750 and accept the order.

Refer to “Using the Web front-end” on page 263 to submit the orders with details stated above. Use the human task application to accept, decline, and ship the orders.

Note: The invocation of the Web service may fail if the port is not 9080. Our code will bypass the Web service and the process continues. You can change the port as described in “Changing the port for the Web service” on page 271.

If you get the error, Connection exception: Name "comp/env/ClipsAndTacks" not found in context "java:" then the data source reference in the ClipsAndTacksF1EJB project was lost, or the JNDI name is missing.

Monitor Dashboard in the unit test environment

Once the three orders are processed, open the dashboard:

- ▶ Select the Monitor Server and *WBM Web Dashboard*.
- ▶ Select the *ClipsAndTacksF1BMP* model and the monitoring contexts are displayed (Figure 12-85).

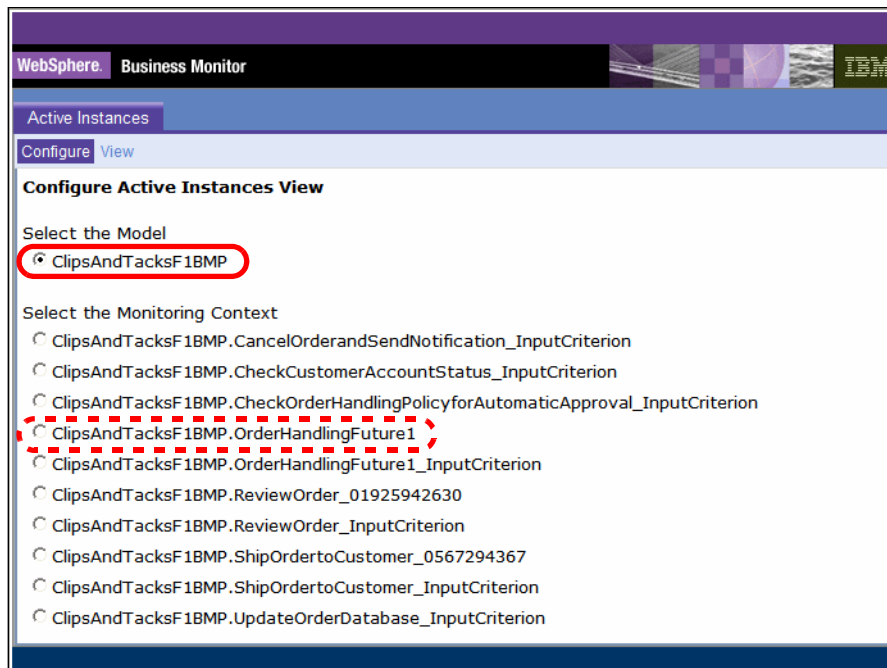


Figure 12-85 Monitor unit test environment (UTE): Select the context

- ▶ Select the monitoring context, then select the metrics to be displayed (Figure 12-86).

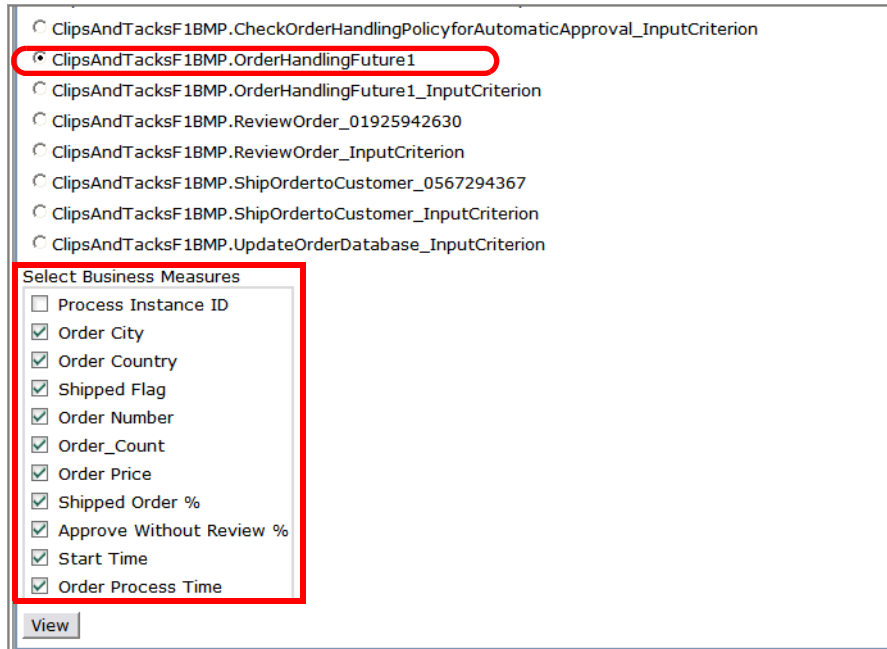


Figure 12-86 Monitor UTE: Select the metrics to be displayed

Active instances

- ▶ Click *View* and the results of the monitor model are displayed (Figure 12-87).

WebSphere Business Monitor									
Active Instances									
Active Instances Values									
Order City	Order Country	Shipped Flag	Order Number	Order_Count	Order Price	Shipped Order %	Approve Without Review %	Start Time	Order Process Time
Etobicoke	Canada	SHIPPED	4056	1	878	100	0	Mar 25, 2007 10:06:35 AM	0d, 0h, 0m, 38s, 594ms
Buffalo	USA	SHIPPED	6441	1	478	100	100	Mar 25, 2007 10:06:35 AM	0d, 0h, 0m, 34s, 656ms
Markham	Canada	NOT SHIPPED	9377	1	1,355	0	0	Mar 25, 2007 10:06:35 AM	0d, 0h, 0m, 28s, 828ms

Figure 12-87 Monitor UTE: Active instances

Note: In the test environment we cannot see aggregated measures over multiple instances. We can only test if the instances produce the underlying data that will be accumulated by the aggregate measures.

- ▶ Submit two more orders and let these orders wait for human interaction.
- ▶ Click *Refresh* in the dashboard to see the new orders. These orders are displayed with the flag NOT_SHIPPED. There is no indication that the orders have not been processed. We have to explore the ShipOrderToCustomer and CancelOrderandSendNotification monitoring contexts to see the shipped and declined order counts.
- ▶ Click *Configure* in the dashboard and select one of the activities and view the instances.

Activity processing time

For some of the activities, you can see the processing time that we measured:

- ▶ Select *ClipsAndTacksF1BMP.CheckOrderHandlingPolicyforAutomatic...* and you can select the active instance ID and the *Check Policy Processing Time* (Figure 12-88 top).
- ▶ Select *ClipsAndTacksF1BMP.UpdateOrderDatabase_InputCriterion* and you can select the *Order Shipped Time*, *Fulfillment Time*, and *Update Order Database Processing Time* (Figure 12-88 bottom).

The figure consists of two screenshots of the Monitor UTE interface. The top screenshot shows the 'Active Instances Values' for the activity 'ClipsAndTacksF1BMP.CheckOrderHandlingPolicyforAutomatic...'. It displays a table with two columns: 'Activity Instance ID' and 'Check Policy Processing Time'. The bottom screenshot shows the 'Active Instances Values' for the activity 'ClipsAndTacksF1BMP.UpdateOrderDatabase_InputCriterion'. It displays a table with four columns: 'Activity Instance ID', 'Order Shipped Time', 'Fulfillment Time', and 'Update Order Database Processing Time'.

Active Instances Values	
Activity Instance ID	Check Policy Processing Time
_AI:90040111.8a106f9e.99fe573f.170a0041	0d, 0h, 0m, 5s, 547ms
_AI:90040111.8a106fcc.99fe573f.170a0043	0d, 0h, 0m, 5s, 562ms
_AI:90040111.8a106fbd.99fe573f.170a0042	0d, 0h, 0m, 5s, 547ms

Active Instances Values			
Activity Instance ID	Order Shipped Time	Fulfillment Time	Update Order Database Processing Time
_AI:90040111.8a10d1e2.99fe573f.170a0098			0d, 0h, 0m, 0s, 140ms
_AI:90040111.8a10e098.99fe573f.170a009c	Mar 25, 2007 10:07:13 AM	38109	0d, 0h, 0m, 0s, 93ms

Figure 12-88 Monitor UTE: Processing time metrics for activities

- ▶ Select the ReviewOrder_XXXXX monitoring context. The only data that can be viewed is the processing time (Figure 12-89).

Active Instances Values	
Human Task Instance ID	Review Order Duration
_TKI:a01b0111.8a109323.99fe573f.170a005a	0d, 0h, 0m, 0s, 250ms
_TKI:a01b0111.8a10990f.99fe573f.170a0066	0d, 0h, 0m, 0s, 110ms

Figure 12-89 Monitor UTE: Review order processing time

Exporting the monitor model EAR file

Once the monitor model is created and tested, we have to deploy the model on the WebSphere Monitor Server:

- ▶ Switch to Web perspective.
- ▶ Select *ClipsAndTacksF1BMPEAR* → *Export* → *EAR file*.
- ▶ Specify the destination:
 - C:\SG247148\sampcode\monitor\earfiles\ClipsAndTacksF1BMPEAR.ear
- ▶ Click *Finish*.

Undeploying the monitor model in the test environment

You can only test one monitor model at a time. To remove the current model from the test environment, perform these steps:

- ▶ Start the administrative console and login.
- ▶ Select *Applications* → *Monitor Models*, then select the applicable model and click *Stop*.
- ▶ Open a DB2 command window and navigate to the BMPScripts directory containing the output produced by data services generation when the version was first installed.
- ▶ In a text editor (WordPad), open the stateDrop.d11 script and uncomment the drop table statements (Figure 12-90).

```
-- WARNING - EXECUTING THE CONTENTS OF THIS SCRIPT WILL DROP YOUR DATABASE
-- TABLES, CAUSING THE IRREVERSIBLE LOSS OF ANY DATA WHICH IS NOT BACKED UP.
-- IN ORDER TO EXECUTE THIS SCRIPT YOU MUST FIRST MANUALLY UNCOMMENT THE SQL
-- COMMANDS BELOW.

ECHO YOU MUST EDIT THIS SCRIPT AND UNCOMMENT THE COMMANDS BEFORE RUNNING IT;
--- drop table WBI.CTX_ORDERHANDLINGFUTURE1;
--- drop table WBI.CTX_ORDERHANDLINGFUTR1NPTCRTRN;
--- drop table WBI.CTX_CHKCRDRHNDLNGPLCYFRIMTCPPR;
--- drop table WBI.CTX_CHECKCSTMRCNNTSTTSNPTCRTRN;
--- drop table WBI.CTX_SHIPORDERTOCUSTOMRNPTCRTRN;
--- drop table WBI.CTX_UPDATEORDERDATABASNPTCRTRN;
--- drop table WBI.CTX_REVIEWORDER_INPUTCRITERION;
--- drop table WBI.CTX_CNCLDRDNRDNDNTFCINNPTCRTRN;
--- drop table WBI.CTX_REVIEWORDER_01925942630;
--- drop table WBI.CTX_SHIPORDERTOCSTMRO567294367;
--- delete from REPOS.DBSETUP where MODELID='ClipsAndTacksF1BMP';
```

Figure 12-90 StateDrop.ddl file for undeployment

- ▶ Run the following commands to remove the monitor model schema for all versions of the specified model:

```
db2 connect to monitor
db2 -tvf stateDrop.ddl
db2 commit
db2 disconnect monitor
```
- ▶ In the Servers view select the WebSphere Business Monitor Server v6.0.2 and *Add and remove projects*. Then remove the monitor model application (or all the applications) from the server.
- ▶ Stop the Monitor Server.

Summary

In this chapter we described in detail how to implement the monitoring model using WebSphere Integration Developer and the Monitor Toolkit.

We showed how to implement measures, dimensions, KPIs, and alerts. Then we deployed the necessary configurations required to be able to achieve the round trip. Finally, we tested the developed Monitor model in the Monitor test environment and used the Monitor Dashboard to view the instances.



Part 4

Deployment and monitoring

In Part 4 we describe how to deploy the application to WebSphere Process Server and how to measure the application using WebSphere Business Monitor.

We describe in detail how to deploy the business measures developed with the Monitor Toolkit to the production Monitor Server.



Deploying and running the application in Process Server

In this chapter we describe how to export the finished application from WebSphere Integration Developer and deploy it into a production ready WebSphere Process Server.

We describe how to configure the server with a data source and show you how to install the enterprise application.

Then we describe how to run the application using the Web front-end, the BPC Explorer, and the human task application.

Note: The activities in this chapter are performed by the system administrator.

Exporting the application from Integration Developer

The complete ClipsAndTacks application consists of three enterprise applications:

- ▶ ClipsAndTacksF1App—Business process
- ▶ ClipsAndTacksFxHumanCustomEAR—Human task application
- ▶ CreditServiceEAR—Web service

We have to install all three applications in the WebSphere Process Server.

Preparation

Open the OrderHandlingFuture1 process and in the Properties view, Details, select *Automatically delete the process after completion* (important for the production environment, see Figure 10-19 on page 227).

Web service deployment

In the test environment, the Web service is invoked with the URL:

```
http://localhost:9080/CreditServiceWeb/services/CreditBean
```

This binding is visible in the Properties view (Binding tab) when you select the Check Credit Import from the assembly diagram.

In a real process server, this URL might not work and would have to be changed to a real Web address, for example:

```
http://www.clipsandtacks.com/CreditServiceWeb/services/CreditBean
```

For our scenario, you can leave the localhost:9080 address, unless the server uses a different set of ports, for example, localhost:9081.

Tip: For testing and measuring in the process server, you can also reset the switch in the CreditRating bean:

```
static boolean useWebservice = false;
```

See “Implementing the credit check” on page 234 and “Invoking the Web service from an activity” on page 269. With the switch set to false, you do not have to deploy the Web service application to the Process Server.

Exporting the EAR files

To create the EAR files for the process server, select each enterprise application and *Export* → *EAR File* (context):

- ▶ Click *Browse* to enter a destination (Figure 13-1), for example,

C:\SG247148\sampcode\wps\ClipsAndTacksF1App.ear

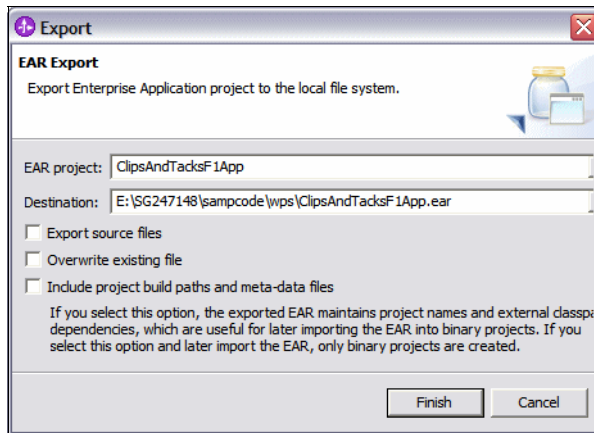


Figure 13-1 Exporting an enterprise application

- ▶ Click *Finish*.
- ▶ Repeat this for every enterprise application.

Copy the EAR file to the system where WebSphere Process Server is installed, for example into the `installableApps` directory:

```
<WPS-HOME>\profiles\ProcSrv01\installableApps
```

Important: Before exporting, decide what security you have configured in Process Server. The application that you export must match the target security. Configure the human tasks to use the security of the target server, that is, set the JNDI name of the staff plug-in configuration in the Properties view Details tab.

Configuring the Process Server

The process server must be configured with the CLIPTACK database that is used to store the orders. We have to define the same data source (`jdbc/cliptack`) that we defined in the test environment (see “Configuring the Process Server test environment” on page 215).

We can use the administrative console as described in “Creating a data source for the database” on page 218, or we can use a JAAS script.

Using a JACL script to define the data source

We can automate configuration definitions by using JACL scripts. To define the data source, follow these steps:

- ▶ Start the process server if it is not running.
- ▶ Open a command window in the process server BIN directory:

```
cd <WPS-HOME>\profiles\ProcSrv01\bin
```
- ▶ Copy the JACL script into the same directory from:

```
SG247148\sampcode\cloudscape\datasource\DatasourceCLIPTACK.jacl
```
- ▶ Edit the `DatasourceCLIPTACK.jacl` file and change the variable `dblocation` to point to the server installation directory:

```
set dblocation "C:/<WPS-HOME>/profiles/ProcSrv01"
```
- ▶ Run the JACL script using the command:

```
wsadmin.bat -f DatasourceCLIPTACK.jacl
```

Verify the data source and create the database

To verify the data source definition in the server, open the administrative console:

- ▶ Select *Resources* → *JDBC Providers*.
- ▶ Select *Server* and click *Apply*. You should see a new JDBC Provider named `Cloudscape JDBC Provider ClipsAndTacks (XA)`.
- ▶ Click *Cloudscape JDBC Provider ClipsAndTacks (XA)*, then click *Data sources*. You should see the `ClipsAndTacks` data source.
- ▶ Click *ClipsAndTacks* and verify the settings of the data source.
- ▶ Click *Custom properties* (right side) and verify that `createDatabase` is set to `create`.
- ▶ In the Data sources list, select the *ClipsAndTacks* data source (check box) and click *Test connection*. This action creates the database and you should receive a successful message.

Configure security in Process Server

Configure security using LDAP in the server as described in “Enabling security for the Process Server” on page 603.

Alternatively you can use the simple user registry for security as described in “Using a custom user registry for security” on page 304. You have to place the files into `<WPS-HOME>\profiles\ProcSrv01`.

Installing the application in Process Server

Enterprise application are usually installed using the administrative console. We use the same steps for every EAR file to install the application in the server:

- ▶ Expand *Applications* and select *Install New Application*. The application installation dialog starts.
- ▶ Click *Browse* to locate the EAR file (Figure 13-2):
`<WPS-HOME>\profiles\ProcSrv01\installableApps\ClipsAndTacksAppF1.ear`

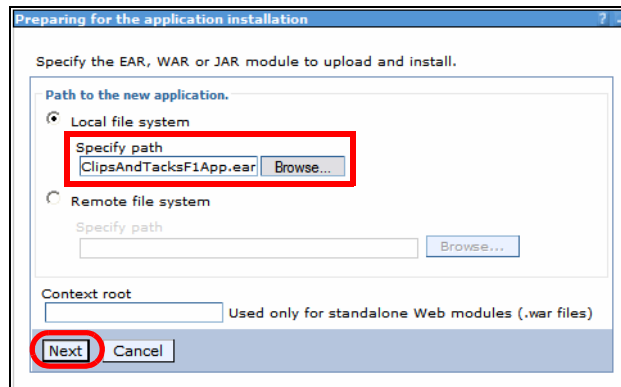


Figure 13-2 EAR installation: Location

- ▶ In the subsequent panels, we basically accept the defaults:
 - Prepare for the application installation.
 - Step 1: Select installation options (select *Pre-compile JSP*).
 - Step 2: Map modules to servers.
 - Step 3: Provide listener bindings for message-driven beans.
 - Step 4: Provide JNDI names for Beans.
 - Step 5: Map resource references to resources.
 - Step 6: Map virtual hosts for Web modules.
 - Step 7: Ensure that all unprotected 2.x methods have the correct level of protection.
 - Step 8: Summary.
- ▶ Click *Finish* and the application is installed. Wait for the message:
Application ClipsAndTacksF1App successfully installed.
- ▶ Click *Save to Master Configuration*.

Repeat the application installation for the other enterprise applications, **ClipsAndTacksFxHumanCustomEAR** and **CreditServiceEAR**. Note that the number of steps is smaller for these applications.

Start the applications

After installation, the applications can be started:

- ▶ Select *Applications* → *Enterprise Applications*.
- ▶ Select the three new applications and click *Start*.

Running the application (without monitor)

We are ready to run the application. However, first we have to populate the CLIPTACK database for the front-end Web application.

We did not install an HTTP server, so we have to submit all requests to **localhost:9080** (or the port of the server profile).

Creating the database

To create the CLIPTACK database, open a browser and submit the URL:

```
http://localhost:9080/ClipsAndTacksF1Invoke/CreateDatabaseServlet
```

Then run the list servlet to verify the database content:

```
http://localhost:9080/ClipsAndTacksF1Invoke/ListServlet
```

Running the Web front-end to submit an order

Start the Web front-end with the URL:

```
http://localhost:9080/ClipsAndTacksF1Invoke/
```

Login as a customer and submit an order over \$750. Refer to Figure 10-54 on page 263 and subsequent figures for instructions.

Using the BPC Explorer

Start the BPC Explorer using the URL:

```
http://localhost:9080/bpc
```

Refer to “Using the human task JSPs with the BPC Explorer” on page 265.

- ▶ You should see the Review Order task waiting for human processing (refer to Figure 10-57 on page 265 and subsequent figures).
- ▶ Select the task and click *Work on*.
- ▶ Select *Approve* and click *Complete*.
- ▶ Refresh the task list and the order is ready for shipment.
- ▶ Select the Ship Order to Customer task and click *Work on*.
- ▶ Click *Complete*.
- ▶ Submit more orders.

Using the customized human task application

Refer to “Using the human task application” on page 284 for instructions.

Submit another order (over \$750) using the Web front-end, then:

- ▶ Start the human task application using the URL:
`http://localhost:9080/ClipsAndTacksFxHumanCustomWeb`
- ▶ Click *Review Orders*, then approve the order (refer to Figure 10-73 on page 284 and subsequent figures).
- ▶ Click *Shipments* and ship the order.
- ▶ Submit an order under \$750 to verify that automatic approval works. If you installed the Web service, it will be invoked.

Using the Business Rules Manager

The Business Rules Manager is not installed in the stand-alone Process Server. We have to perform an installation task:

- ▶ Open a command window in `<WPS-HOME>\profiles\ProcServ01\bin`.
- ▶ Run the JACL script to install the Business Rules Manager:
`wsadmin.bat -f installBRManager.jacl`
- ▶ The Business Rules Manager is installed and started. If you open the administrative console, you can find the `BusinessRulesManager` application in the list of enterprise applications.

To run the Business Rules Manager on the Process Server, open a browser and enter this URL:

`http://localhost:9080/br/webclient`

Refer to “Using the Business Rules Manager” on page 295 for instructions.

Accessing the server log

The log of the Process Server can be found in:

```
<WPS-HOME>\profiles\ProcSrv01\logs\server1\SystemOut.log
```

Using the Common Base Event Browser

The Common Base Event Browser can be used to browse the events that are generated by the application.

Note that this only works if you have configured the process and the activities to emit events, as described in “Generating CEI events for BPEL elements” on page 334.

After submitting and processing a few orders, use the Common Base Event Browser:

- ▶ Open the administrative console.
- ▶ Select *Integration Applications* → *Common Base Event Browser* (Figure 13-3).

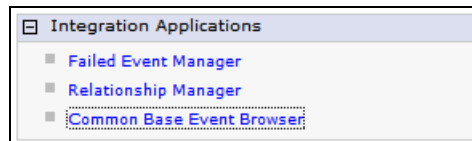


Figure 13-3 Common Base Event Browser: Start

- ▶ Click *Get Events* (at the top). Limit the number of events retrieved (default is 500) and possibly use a start and end time. Click *Get Events* (at the bottom). The number of events retrieved is displayed (Figure 13-4).

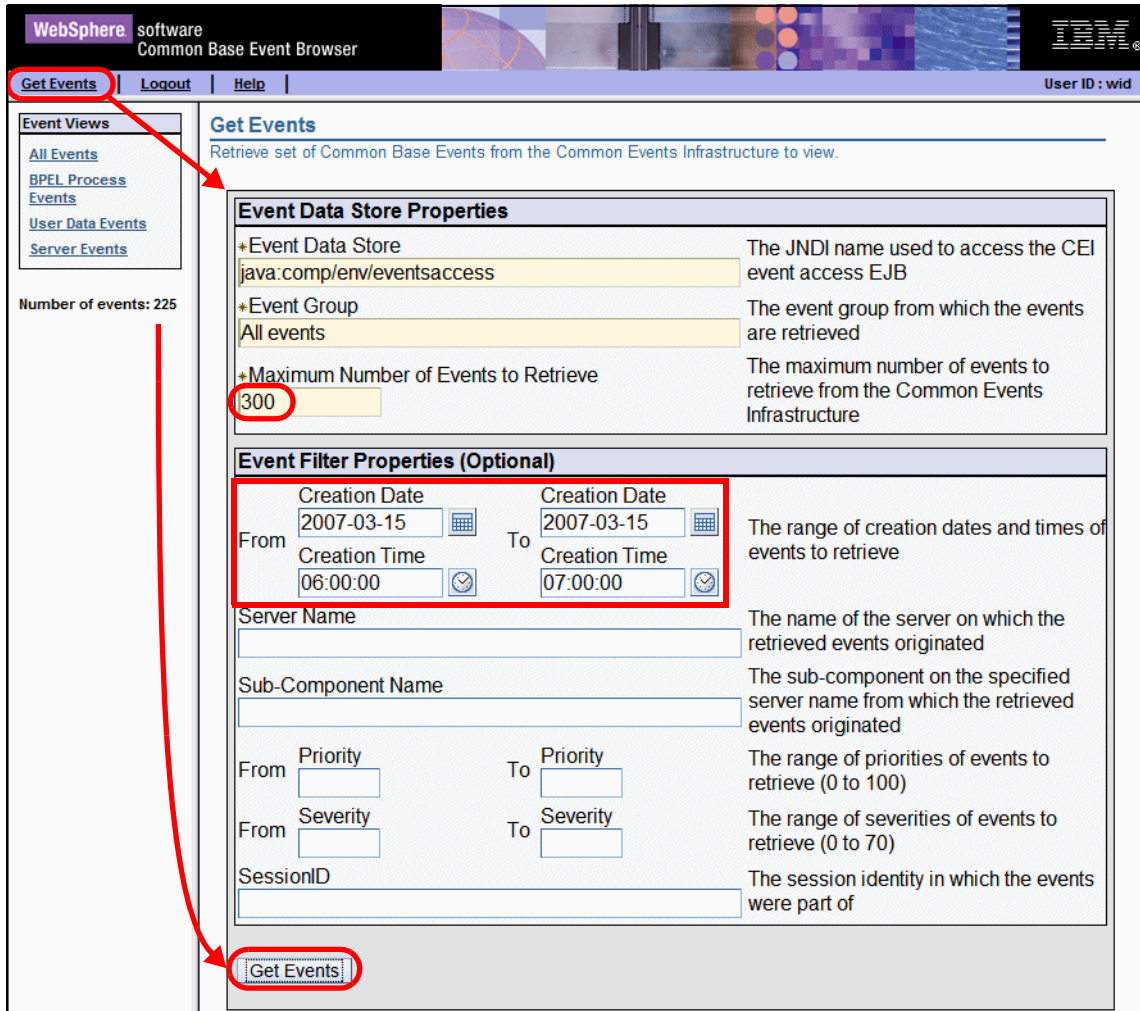


Figure 13-4 Common Base Event Browser: Get Events

- ▶ Click *All Events* in the Events Views box to see a listing of the events. Page through the listing. Select an event to see the details with the application, process, and activity names (Figure 13-5).

Select	Creation Time	Name	Priority
<input type="radio"/>	2007-03-15T06:07:51.203Z	BPC.BFM.PROCESS.STATUS	
<input type="radio"/>	2007-03-15T06:07:51.375Z	BPC.BFM.PROCESS.STATUS	
<input type="radio"/>	2007-03-15T06:09:33.844Z	BPC.BFM.PROCESS.START	
<input type="radio"/>	2007-03-15T06:09:33.875Z	BPC.BFM.VARIABLE.STATUS#yAKf/4BGi351c2jHookCQ	
<input type="radio"/>	2007-03-15T06:09:33.953Z	BPC.BFM.LINK.STATUS	
<input type="radio"/>	2007-03-15T06:09:34.406Z	BPC.BFM.ACTIVITY.MESSAGE#SEJ1+g7UFxtwcAvSreeL8A	
<input type="radio"/>	2007-03-15T06:09:34.750Z	BPC.BFM.VARIABLE.STATUS#yAKf/4BGi351c2jHookCQ	
<input checked="" type="radio"/>	2007-03-15T06:09:34.797Z	BPC.BFM.ACTIVITY.MESSAGE#SEJ1+g7UFxtwcAvSreeL8A	
<input type="radio"/>	2007-03-15T06:09:34.875Z	BPC.BFM.LINK.STATUS	
<input type="radio"/>	2007-03-15T06:09:34.875Z	BPC.BFM.LINK.STATUS	

Page 13 of 23 13 Go Total: 225 Filtered: 225 Displayed: 10

Event Data

List of all properties associated with the selected event.

Name	Value
version	1.0.1
globalInstanceid	CEE1A6FC01DB5481FAA1DBD28BBF4FFFD0
extensionName	BPC.BFM.ACTIVITY.MESSAGE#SEJ1+g7UFxtwcAvSreeL8A
localInstanceid	
creationTime	2007-03-15T06:09:34.797Z
severity	10
msg	
priority	
sequenceNumber	128
repeatCount	
elapsedTime	
contextDataElement / WBSESSION_ID / contextValue	9.43.32.121.ClipsAndTacksF1.processes/orderhandlingfuture1/O
contextDataElement / ECSCurrentID / contextValue	Af90040111.54372ec8.86dfd4f6.6d990136
contextDataElement / ECSParentID / contextValue	Pf90030111.54372c36.86dfd4f6.6d990120
extendedDataElement / PayloadType	full
extendedDataElement / EventNature	EXIT
extendedDataElement / BPCEventCode	21011
extendedDataElement / processTemplateName	OrderHandlingFuture1
extendedDataElement / processTemplateValidFrom	Tue 2007-02-06 21:25:41.000
extendedDataElement / activityKind	21 - KIND_INVOKE
extendedDataElement / state	5 - STATE_FINISHED
extendedDataElement / bpelId	1006
extendedDataElement / activityTemplateName	CheckOrderHandlingPolicyforAutomaticApproval_inputCriterion

Figure 13-5 Common Base Event Browser: List of events and details

- Scroll down further in the details, where you can find the process data (Figure 13-6).

extendedDataElement / activityTemplateName	CheckOrderHandlingPolicyforAutomaticApproval_InputCriteria
extendedDataElement / activityTemplateId	_AT:90020111.2e817b9b.86dfd4f6.843e0126
extendedDataElement / message	
extendedDataElement / message / Output	
extendedDataElement / message / Output / Customer	
extendedDataElement / message / Output / Customer / CustomerNumber	11111
extendedDataElement / message / Output / Customer / CompanyName	Auto Insurance Company
extendedDataElement / message / Output / Customer / ContactFirstName	Richard
extendedDataElement / message / Output / Customer / ContactLastName	Doe
extendedDataElement / message / Output / Customer / StreetAddress	27 New Street
extendedDataElement / message / Output / Customer / City	Etobicoke
extendedDataElement / message / Output / Customer / Country	Canada
extendedDataElement / message / Output / Customer / PostalCode	M8Z 2S6
extendedDataElement / message / Output / Customer / Email	richard@autoinsurance.com
extendedDataElement / message / Output / Customer / Rating	696
extendedDataElement / message / Output / Customer / AvailableCredit	1302.99
extendedDataElement / message / Output / OrderItems	
extendedDataElement / message / Output / OrderItems / ProductName	All-In-One Printer
extendedDataElement / message / Output / OrderItems / ProductNumber	RB-0001
extendedDataElement / message / Output / OrderItems / Price	150.0
extendedDataElement / message / Output / OrderItems / Description	
extendedDataElement / message / Output / OrderItems / Quantity	1
extendedDataElement / message / Output / TotalPrice	150.0
extendedDataElement / message / Output / OrderNumber	6467
extendedDataElement / message / Output / OrderStatus	NEW
extendedDataElement / message / Output / ProcessingPreference	
extendedDataElement / message / Output / ProcessingPreference / automaticApproval	true

Figure 13-6 Common Base Event Browser: Detailed process data

- ▶ Click *BPEL Process Events* in the Events Views box, then select the process and an instance to see the events of one process instance (Figure 13-7).

Process	Instance	Select	Creation Time	Extension Name
OrderHandlingFuture2	PI:90030111.53b82e82.86dfd4f6.3	<input type="radio"/>	2007-03-15T05:00:52.031Z	BPC.BFM.ACTIVITY.MESSAGE#C
OrderHandlingFuture1	PI:90030111.54372c36.86dfd4f6.f	<input type="radio"/>	2007-03-15T05:00:52.062Z	BPC.BFM.LINK.STATUS
		<input type="radio"/>	2007-03-15T05:00:52.250Z	BPC.BFM.ACTIVITY.STATUS
		<input checked="" type="radio"/>	2007-03-15T05:00:52.281Z	BPC.BFM.ACTIVITY.STATUS
		<input type="radio"/>	2007-03-15T05:00:52.328Z	BPC.BFM.PROCESS.STATUS
		<input type="radio"/>	2007-03-15T05:00:52.469Z	BPC.BFM.PROCESS.STATUS
Page 1 of 1			Total: 300 Filtered: 6	

Figure 13-7 Common Base Event Browser: Instances by process

The Common Base Event Browser should be used before deploying the monitor model to verify that events are generated by the application.

Uninstalling the application

If you make changes to the application in Integration Developer, you have to reinstall the new EAR file after uninstalling the old application.

Before an old application can be uninstalled, we must make sure that there are no active instances.

We use the BPC Explorer to terminate active task and process instances:

- ▶ Under task instances select *Administered By Me*.
- ▶ Select all the tasks and click *Terminate*.
- ▶ Under process instances select *Administered By Me*.
- ▶ Select all the processes and click *Delete*.

Note: This action is only necessary when you set the flag to keep processes when they are finished (Figure 10-19 on page 227).

Deleting the instances is not necessary in the Integration Developer test environment where active instances are removed automatically.

The we make sure that no new instances can be created:

- ▶ Open the administrative console.
- ▶ Select *Applications* → *Enterprise Applications*, then select the enterprise application (C1ipsAndTacksF1App).
- ▶ Under Related items, click *EJB Modules*, then select the EJB JAR file.
- ▶ Under Additional Properties, select *Business Processes*.
- ▶ Select all processes and click *Stop*.
- ▶ For the EJB JAR file, under Additional Properties, select *Human Tasks*.
- ▶ Select all task templates and click *Stop*.

Now the application can be stopped and uninstalled:

- ▶ In the administrative console, select *Applications* → *Enterprise Applications*.
- ▶ Select the application (C1ipsAndTacksF1App) and click *Stop*.
- ▶ Select the application (C1ipsAndTacksF1App) and click *Uninstall*.
- ▶ Save the configuration changes.

Finally, you can install the new EAR file using the installation process.

Summary

In this chapter we described how to deploy an application from Integration Developer to a real Process Server.

After configuring the server with the necessary data source, we can install the business process application together with the supporting code.



Deploying the monitor model and measuring the Future 1 process

In this chapter we detail the steps required to deploy our monitor model EAR file on the Monitor server. We configure the Adaptive Action Manager, then configure the Monitor Dashboard.

Following a successful deployment of the model, we submit a set of orders over a period of a few days. We then illustrate how WebSphere Business Monitor can be used to take measurements on the ClipsAndTacks Future 1 process. We review the metrics, measures, KPIs, and alerts for our process using the Monitor Dashboard. We demonstrate how the Monitor Dashboard capabilities can help us analyze this data, including the use of the dimensional view to see our business results by location.

Deploying the monitor model

In this section we describe how to take the monitor model EAR file exported in Chapter 12, “Developing and testing the business measures with the Monitor Toolkit” on page 327 and deploy it on the Monitor Server. New in WebSphere Business Monitor V6.0.2, the Monitor has a setup wizard (also referred to as the *Monitor life cycle*) to guide us through the process of deploying a new monitor model, or a new version of an existing model. The setup wizard is available once a monitor model has been installed on the Monitor Server.

Installing the monitor model

To install the monitor model, follow these steps.

- ▶ Start the Monitor Server (*Programs* → *IBM WebSphere* → *Process Server 6.0* → *Profiles* → *wbmonitor* → *Start the server*) or run the `startServer.bat` command:

```
<WPS-HOME>\ProcServer\profiles\wbmonitor\bin\startServer.bat server1
```

- ▶ Start the administrative console for the Monitor Server (*Programs* → *IBM WebSphere* → *Process Server 6.0* → *Profiles* → *wbmonitor* → *Administrative console*) or open a browser at this URL:

```
http://localhost:9062/ibm/console
```

```
https://localhost:9045/ibm/console/logon.jsp (with security)
```

Use the correct user ID and password if you use security (wid in our installation).

Installing the monitor model consists of two major activities, where the second activity consists of a number of steps.

- ▶ Install the enterprise application.
- ▶ Make the application startable by generating the underlying DB2 tables, DB2 replicators, DB2 Cube Views, DB2 Alphablox, and configuring the Process Server where the application runs.

Installing the monitor model application

Select *Applications* → *Monitor Models*. Currently there are no monitor models installed on the Monitor Server. Click *Install* (Figure 14-1).

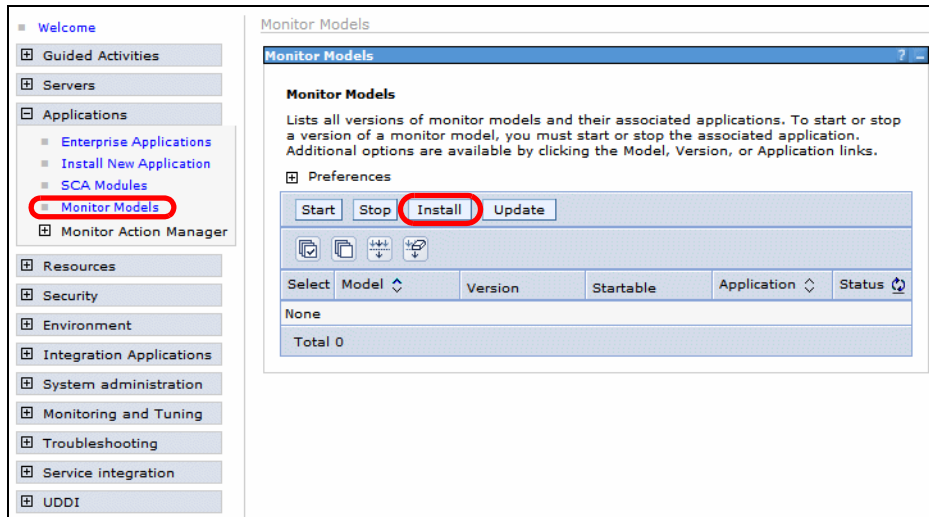


Figure 14-1 Monitor Server administrative console

- ▶ Click *Browse* and select the monitor model EAR file exported in “Deploying the monitor model to the unit test environment” on page 384. Click *Next* (Figure 14-2).

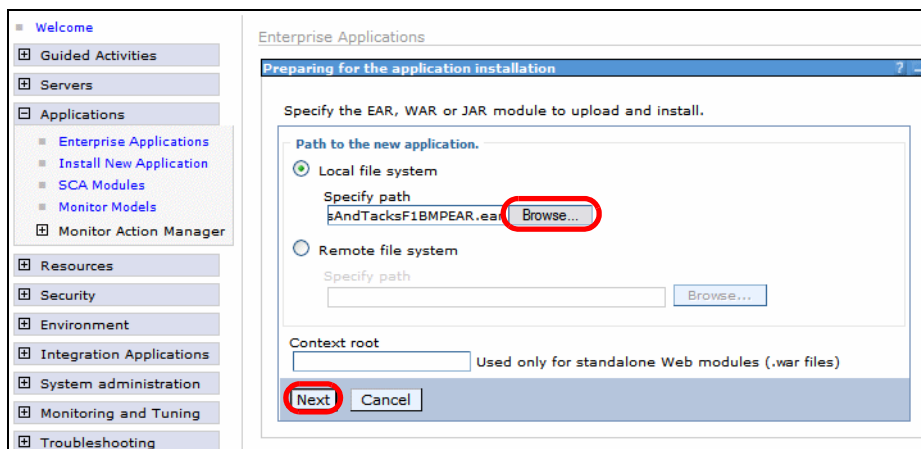


Figure 14-2 Installation of monitor model EAR file

- ▶ On the next page, Preparing for the application installation, we do not overwrite any bindings, so click *Next*.
- ▶ The next page displays security warnings about Java 2 security. Click *Continue*.

- ▶ The next 11 steps are typical steps for installing an enterprise application:
 1. Select installation options.
 2. Map modules to server.
 3. Select current back-end ID.
 4. Provide JNDI names for beans.
 5. Provide default data source mapping.
 6. Map data sources for all 2.x CMP beans.
 7. Map EJB references to beans.
 8. Map resource references to resources.
 9. Map resource env entry references to resources.
 10. Ensure all unprotected 2.x methods have the correct level of protection.
 11. Summary.

We are taking all the defaults. You can go through the steps or click Step 11 to skip to the summary page.

- ▶ Review the installation summary and then click *Finish* (Figure 14-3).



Figure 14-3 Monitor model installation summary

- ▶ Wait while the installation completes. The following message confirms that the installation was successful:

Application ClipsAndTacksF1BMPEAR installed successfully.

- ▶ Click *Save to master configuration* to save the changes (Figure 14-4). Click *Save* on the confirmation panel.

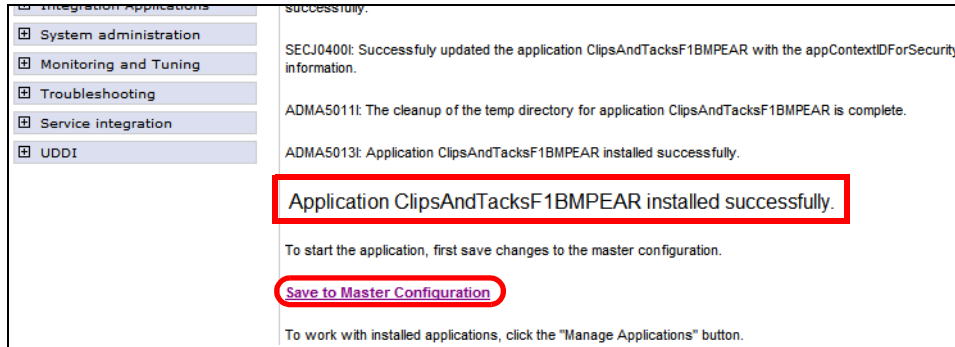


Figure 14-4 Successful installation of the monitor model application

Making the application startable

Once again, select *Applications* → *Monitor Models* from the Administrative Console welcome page. The ClipsAndTacksF1BMP model is now listed, but it is not startable (Figure 14-5).

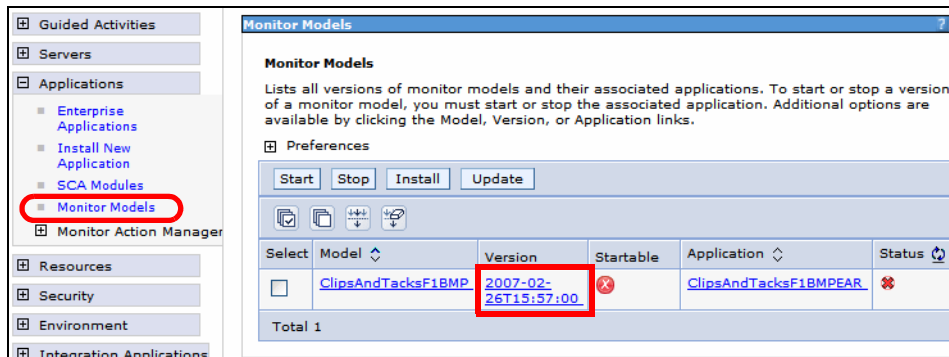


Figure 14-5 Monitor Model prior to executing the setup wizard

Our monitor model requires further configuration before it will be in a startable state. The Monitor model setup wizard guides us through these steps:

- ▶ Click the link in the *Version* column (the date).

- ▶ On the next page (Figure 14-6), the Setup Status in the bottom left of the panel lists the seven steps in the setup wizard. These will be marked green once completed. Underneath Version Properties, select *Setup Wizard* to launch the setup wizard.

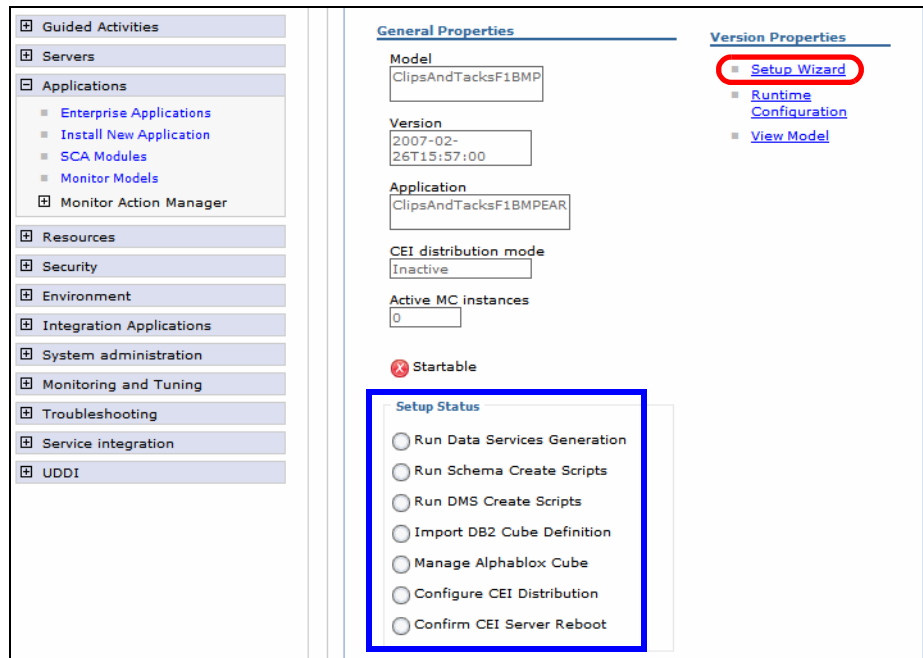


Figure 14-6 Version properties panel

Step 1: Run Data Services Generation

Step 1 of the setup wizard is data services generation. This process generates the artifacts required to complete deployment of the model. A folder identified by the time stamp is created in the specified location when the data services generation step is run. This folder contains the schemas, DB2 Cube Views files, and data movement services scripts required by the setup wizard.

- ▶ Expand the two sections on the panel which defines step 1 of the setup wizard and update these two fields:
 - Output directory: C:\ClipsAndTacksF1
 - Database population interval: 1

The database population interval specifies the frequency of replication between the MONITOR and DATAMART databases.
- ▶ Click *Run* (Figure 14-7).

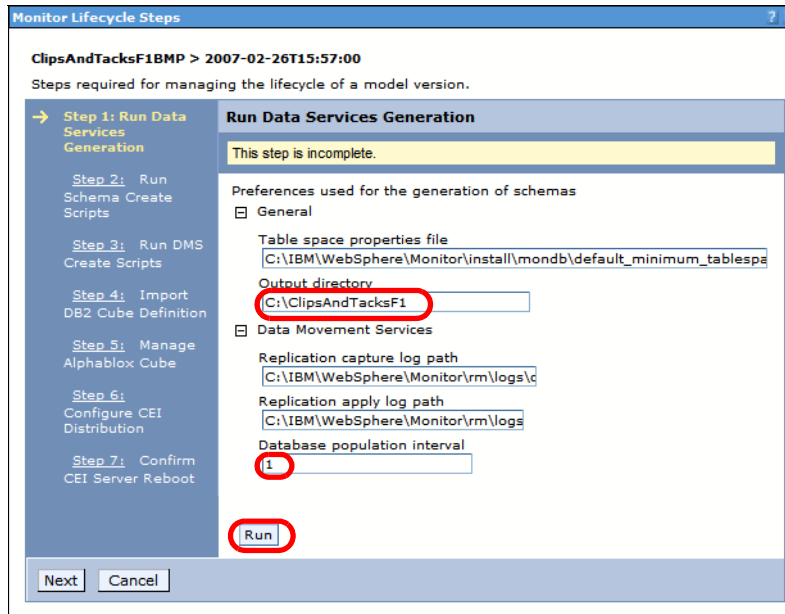


Figure 14-7 Monitor life cycle: Step 1

- Verify that the step completed successfully (Figure 14-8).

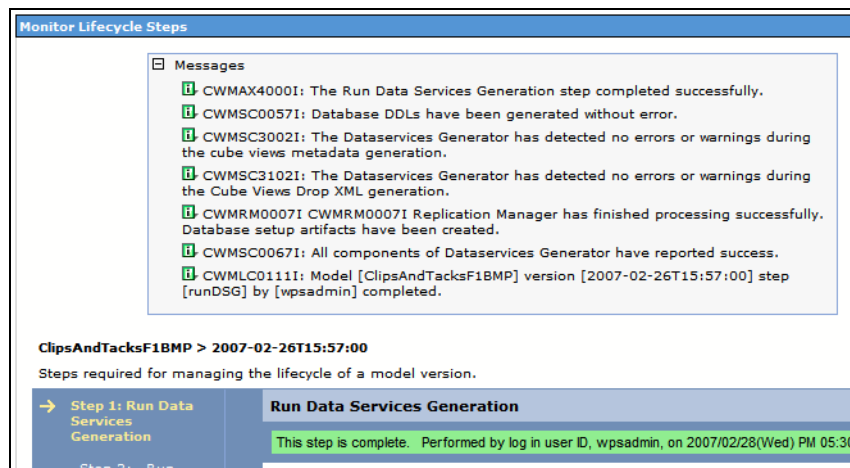


Figure 14-8 Monitor life cycle: Step 1 completed

- Open Windows Explorer to browse the schema folder:
C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx

Three sets of artifacts are produced:

► Schema generation

```
state.ddl
stateDrop.ddl
stateMapping.html
datamart.ddl
datamartDrop.ddl
datamartMapping.ddl
```

► DB2 Cube views generation

```
model_cv.xml
drop_model_cv.xml
import_model.xml
```

► Data movement services

```
DS_DMS_Setup.zip
```

► Locate DS_DMS_setup.zip and unzip it to this folder (Figure 14-9).

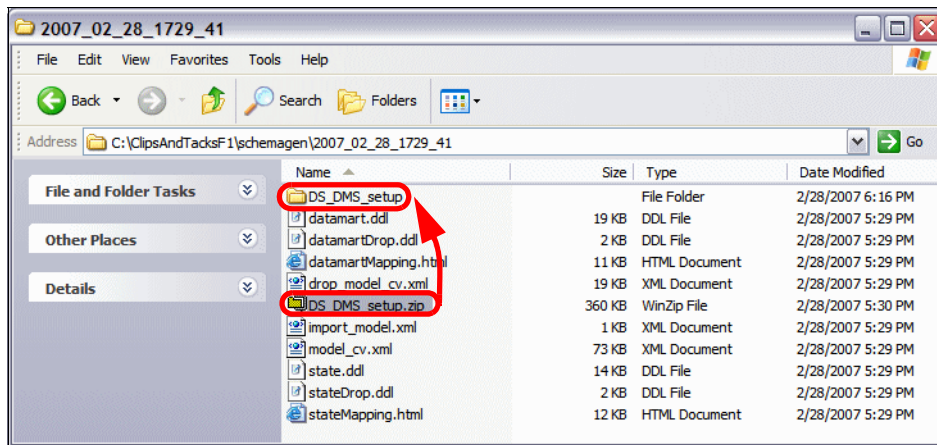


Figure 14-9 Generated schema

Back in the administrative console, click *Next*.

Step 2: Run Schema Create Scripts

Step 2 guides us through the process of running the schema create scripts in a DB2 command window outside of the administrative console. The setup wizard provides details of the commands that have to be run (Figure 14-10).

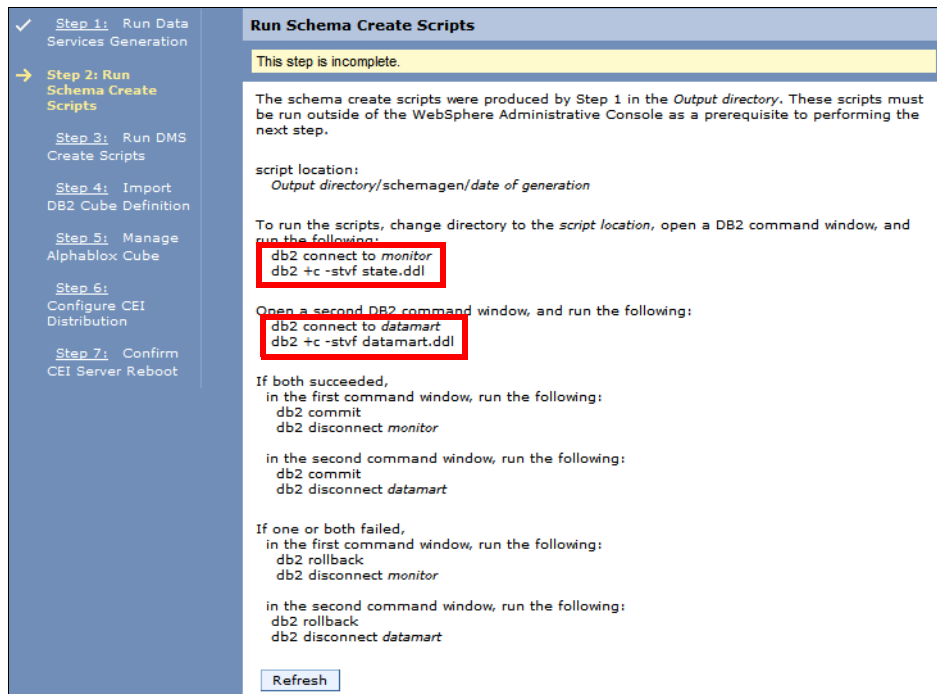


Figure 14-10 Monitor life cycle: Step 2

- ▶ Open a DB2 command window.
- ▶ Change to the directory containing the output of step 1 and run the commands as specified in the setup wizard:


```
cd C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx
db2 connect to monitor
db2 +c -stvf state.ddl
```
- ▶ Open a second DB2 command window and execute the following commands:


```
cd C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41
db2 connect to datamart
db2 +c -stvf datamart.ddl
```
- ▶ Following successful completion of both scripts, commit the updates in each command window:


```
db2 commit
db2 disconnect monitor           or           db2 disconnect datamart
```

Note: If you specified different names for your MONITOR and DATAMART databases, change the commands above accordingly. Refer to the rollback instructions in the setup wizard if you encounter errors running the `state.ddl` or `datamart.ddl` scripts.

- ▶ Back in the administrative console, select *Refresh* and wait for confirmation that the step has completed successfully (Figure 14-11).

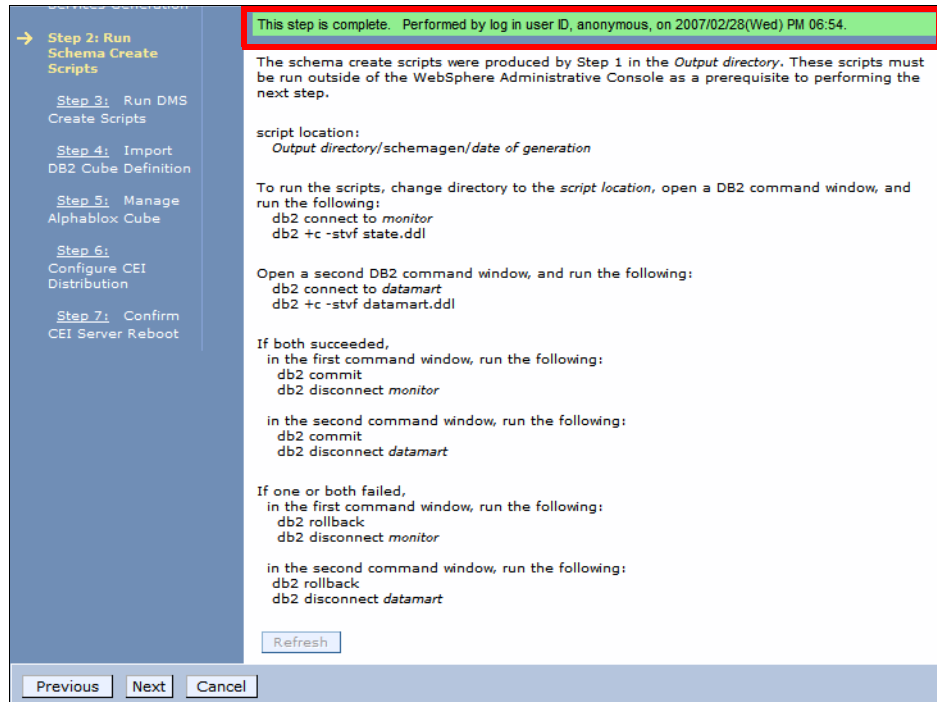


Figure 14-11 Monitor life cycle: Step 2 completed

Step 3: Run DMS Create Scripts

In step 3 we run the data movement services (DMS) scripts to configure the DB2 replication manager to copy metric data from the state tables in the MONITOR database to the star schema tables in the DATAMART database. Follow the instructions in the setup wizard to complete this step:

- ▶ Extract the `DS_DMS_setup.zip` file (we used a `DS_DMS_setup` subdirectory).
- ▶ Open a DB2 command window.

- ▶ Execute the following the commands:

```
cd C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx\DS_DMS_setup
State_to_Datamart_setup_source.bat
```

- ▶ When prompted, enter the DB2 user ID (db2admin) and password (password) to access the MONITOR and DATAMART databases (Figure 14-12).

```
DB2 CLP - State_to_Datamart_setup_source
setup\.\State_to_Datamart\target\CreateStatementRegistration_106.asncli".
CWMMRM9029I Scanning file "C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41\DS_DMS
setup\.\State_to_Datamart\target\CreateStatementRegistration_121.asncli".
CWMMRM9029I Scanning file "C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41\DS_DMS
setup\.\State_to_Datamart\target\CreateStatementRegistration_136.asncli".
CWMMRM9029I Scanning file "C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41\DS_DMS
setup\.\State_to_Datamart\target\CreateStatementRegistration_19.asncli".
CWMMRM9029I Scanning file "C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41\DS_DMS
setup\.\State_to_Datamart\target\CreateStatementRegistration_25.asncli".
CWMMRM9029I Scanning file "C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41\DS_DMS
setup\.\State_to_Datamart\target\CreateStatementRegistration_31.asncli".
CWMMRM9029I Scanning file "C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41\DS_DMS
setup\.\State_to_Datamart\target\CreateStatementRegistration_46.asncli".
CWMMRM9029I Scanning file "C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41\DS_DMS
setup\.\State_to_Datamart\target\CreateStatementRegistration_61.asncli".
CWMMRM9029I Scanning file "C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41\DS_DMS
setup\.\State_to_Datamart\target\CreateStatementRegistration_76.asncli".
CWMMRM9029I Scanning file "C:\ClipsAndTacksF1\schemagen\2007_02_28_1729_41\DS_DMS
setup\.\State_to_Datamart\target\CreateStatementRegistration_91.asncli".
CWMMRM9034I Number of files analyzed: 46. Number of unique database references id
entified: 2.
CWMMRM9019I Type a user ID to be used to connect to database MONITOR:db2admin
CWMMRM9020I Type a password for this user:
CWMMRM9019I Type a user ID to be used to connect to database DATAMART:db2admin
CWMMRM9020I Type a password for this user:
```

Figure 14-12 Execution of State_to_Datamart_setup_source

- ▶ Wait until the script completes, then check the log (State_to_Datamart_setup_source.log) for errors.

Note: The scripts have the prefix *State* as opposed to *Monitor* because only the state tables in the MONITOR database are replicated. The MONITOR database consists of both state and repository tables, and the repository data is not replicated to the DATAMART database.

- ▶ Following successful completion of the State_to_Datamart_setup_source script, run the script to set up the target:


```
State_to_Datamart_setup_target
```
- ▶ Enter the DB2 administrator user ID and password when prompted, then wait for the script to complete.
- ▶ Back in the administrative console, click *Confirm*.
- ▶ Messages should be displayed to indicate successful completion of this step (Figure 14-13).

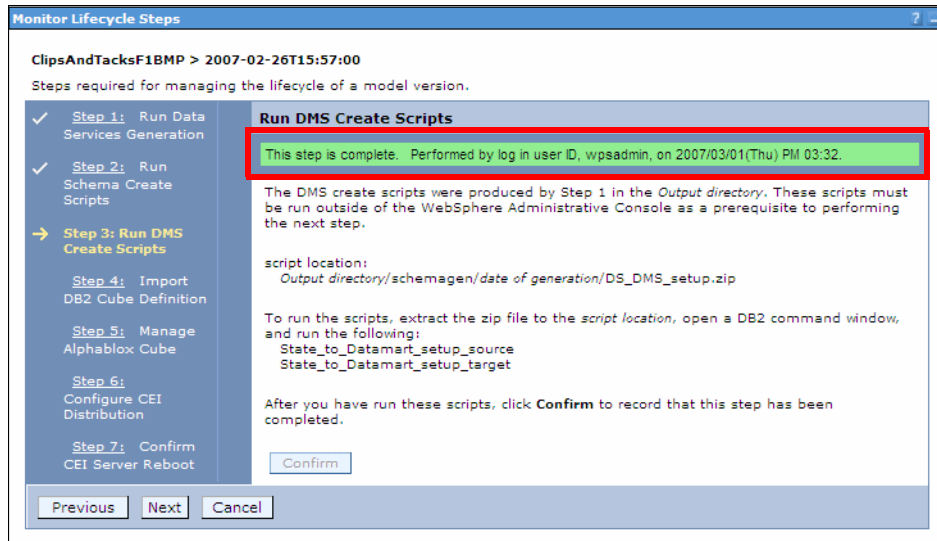


Figure 14-13 Monitor life cycle: Step 3 completed

Step 4: Import DB2 Cube Definition

In step 4 we import the DB2 cube definitions for our monitor model. If you have not previously used your DATAMART database with OLAP (Online Analytical Processing) it is necessary to first configure the DATAMART database to use the DB2 Cube Views feature. Follow the steps outlined in the setup wizard:

- ▶ Open a DB2 command window and execute the following commands:

```
cd c:\IBM\DB2\SQLLIB\MISC
db2 connect to datamart
db2 -tvf db2mdapi.sql
db2 disconnect datamart
```

Note: It is not necessary to execute the step above when deploying subsequent models.

To import the DB2 Cube definitions for our monitor model, execute the following steps, as instructed in the setup wizard:

- ▶ Open a DB2 command window and run the following command:

```
db2mdapiclient -d datamart -i import_model.xml -o response_model_cv.xml -m
model_cv.xml
```

Note: `model_cv.xml` contains all the details regarding the data mart component of the monitoring model developed in the Monitor Development Toolkit.

- ▶ You should receive the following message on completion of the command:

```
====> SQLCODE(0)  SQLSTATE(      ) :
          SQL08026
```
- ▶ Back in the administrative console, click *Confirm*.
- ▶ Verify that the messages indicating successful execution of step 4 are displayed (Figure 14-14).

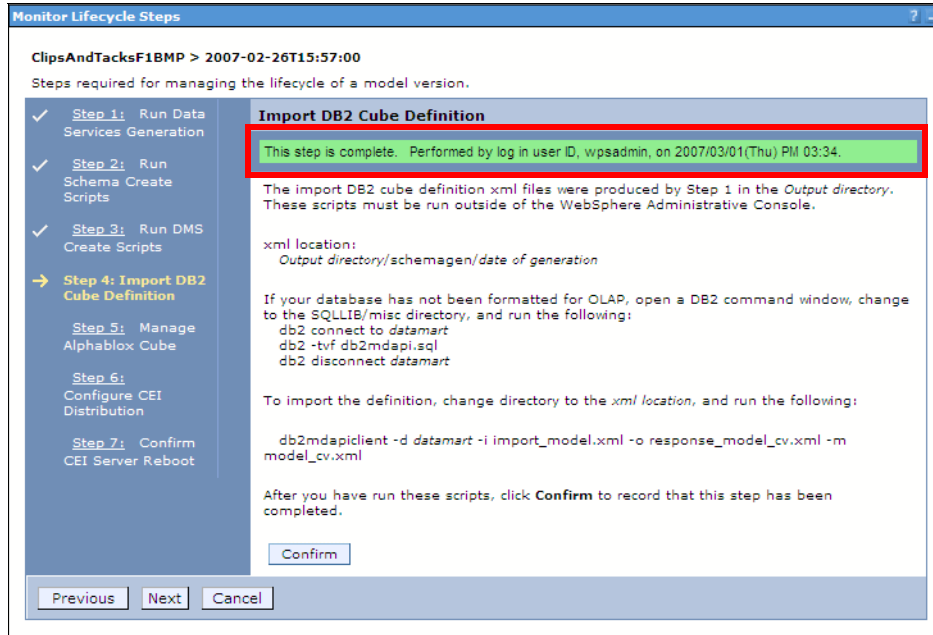


Figure 14-14 Monitor life cycle: Step 4 completed

To understand what has been created by this step, open the DB2 Control Center and select the icon to launch the OLAP Center (Figure 14-15). Alternatively, select *Tools* → *OLAP Center*.

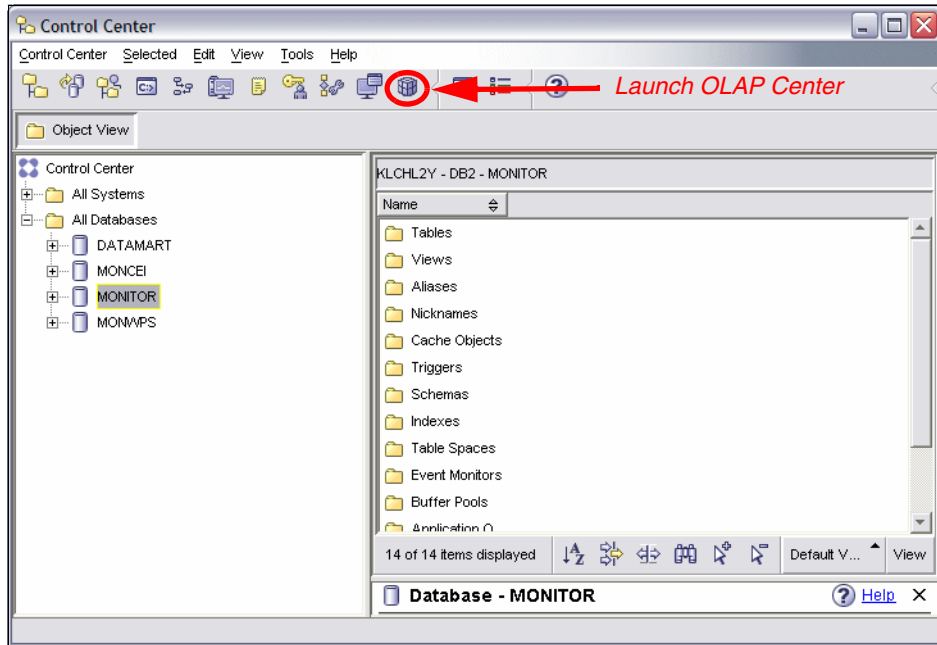


Figure 14-15 DB2 Control Center: Launch OLAP Center

In the OLAP Center, browse the cubes that have been created (Figure 14-16). On the left, we can see that one cube has been created for each monitoring context in the monitor model.

Note: In our model we have a 1:1 relationship between the monitoring contexts and their associated cubes. However, this does not have to be the case, we could have created additional cubes in the Monitoring Model Editor, if desired.

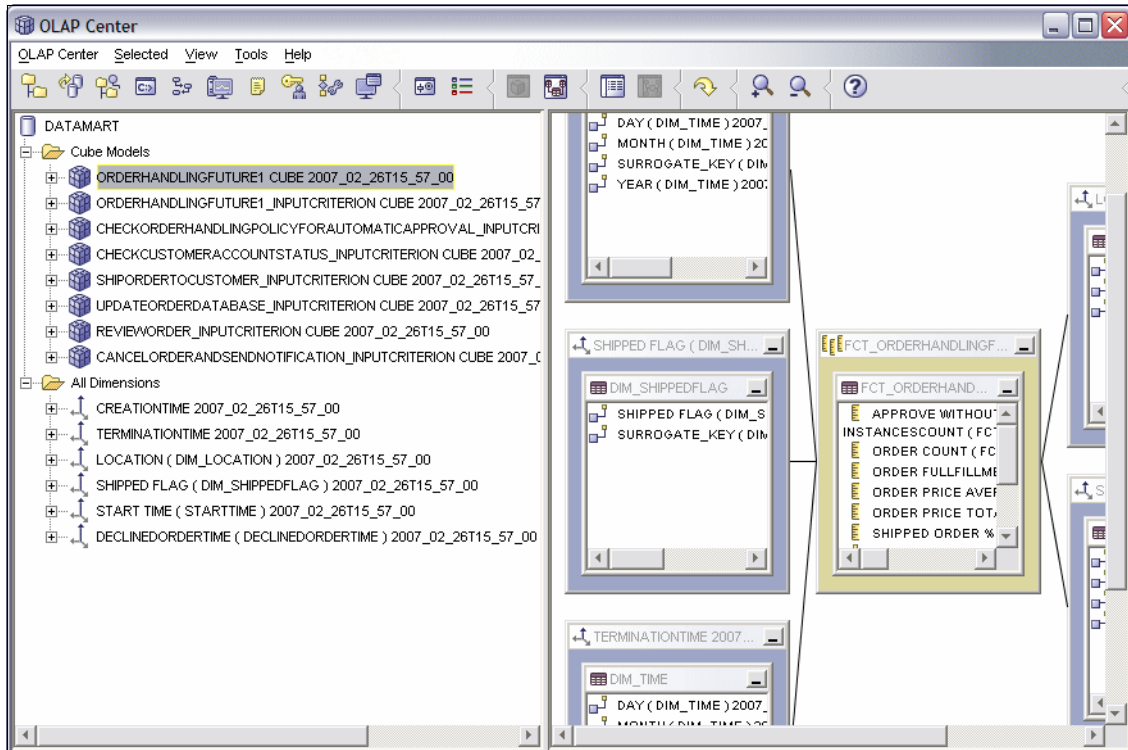


Figure 14-16 OLAP Center

Step 5: Manage Alphablox Cubes

In step 5 we create the Alphablox cubes. DB2 Alphablox provides Web-based analytic applications for relational databases, relational cubes in DB2, and multidimensional databases. It is a powerful tool for manipulating and analyzing the data stored in the DATAMART database. It uses the configuration provided by DB2 Cube Views (step 4) to enhance its performance in executing these queries.

To run step 5, complete the required fields in the setup wizard:

- ▶ **Ensure that the Monitor Dashboard Portal Server is running.**
- ▶ The WebSphere Portal for the Monitor Dashboard is installed on our local system, so we enter the host name (KLCHL2Y) and RMI port number (2810).
- ▶ Select security *enabled* and provide the administrator login details for the WebSphere Application Server used by the Dashboard Portal Server (user ID wpsbind, password wpsbind).
- ▶ Select *Create* to create the Alphablox cubes for the monitor model (Figure 14-17).

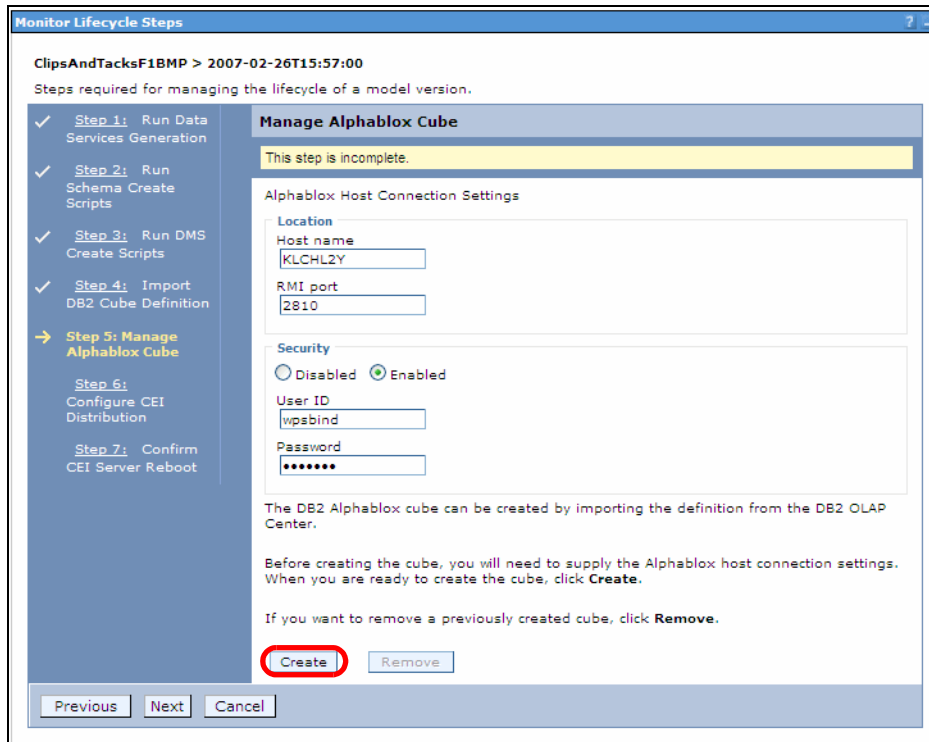


Figure 14-17 Monitor life cycle: Step 5

- ▶ The administrative console will refresh and display messages indicating that this step has completed successfully (Figure 14-18).

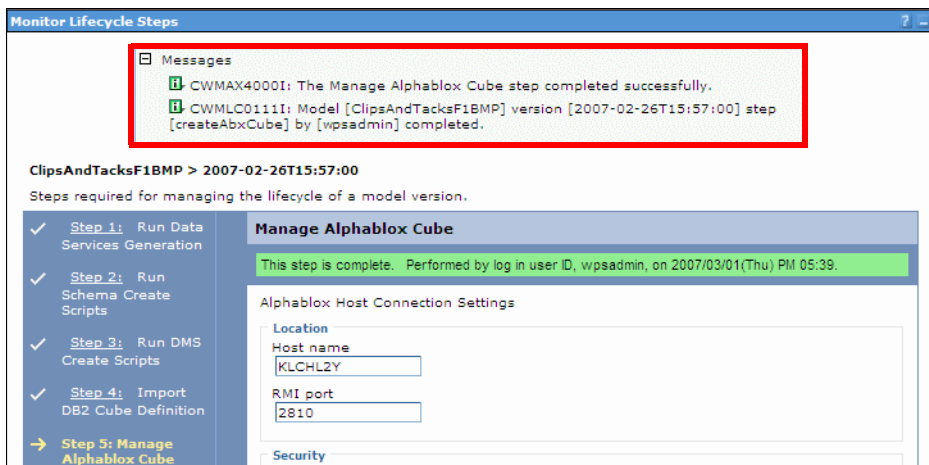


Figure 14-18 Monitor life cycle: Step 5 completed successfully

Step 6: Configure CEI Distribution

In this step the Common Event Infrastructure (CEI) is configured to distribute the events emitted by the business process. Events emitted by the business process running on WebSphere Process Server have to be delivered to the Business Monitor. The event source uses the CEI to deliver these events. The CEI provides filtering of events and distributes events to one or more WebSphere Platform Messaging queue destinations that have been configured for use with the Monitor. First we configure the CEI distribution as detailed in step 6 of the setup wizard. Then we configure the queue destination that is created on the Process Server.

Follow the instructions in the setup wizard:

- ▶ Enter the details for the remote WebSphere Process Server system where the business process is running (Figure 14-19):
 - Location: Remote
 - Host name: `wps.ibm.com` (mapped to the remote system's IP address in `C:\WINDOWS\system32\drivers\etc\hosts`)
 - RMI port: 2809
 - Cell: `wpsNode01Cell`

The WebSphere Process Server cell name was determined as follows:

 - Launch the administrative console for WebSphere Process Server.
 - Select *Resources* → *JMS Providers* → *Default messaging*.
 - Note the cell name: `wpsNode01Cell`
 - Security: enabled
 - User ID: `wid`
 - Password: `wid`
- ▶ Click *Apply* (Figure 14-19).

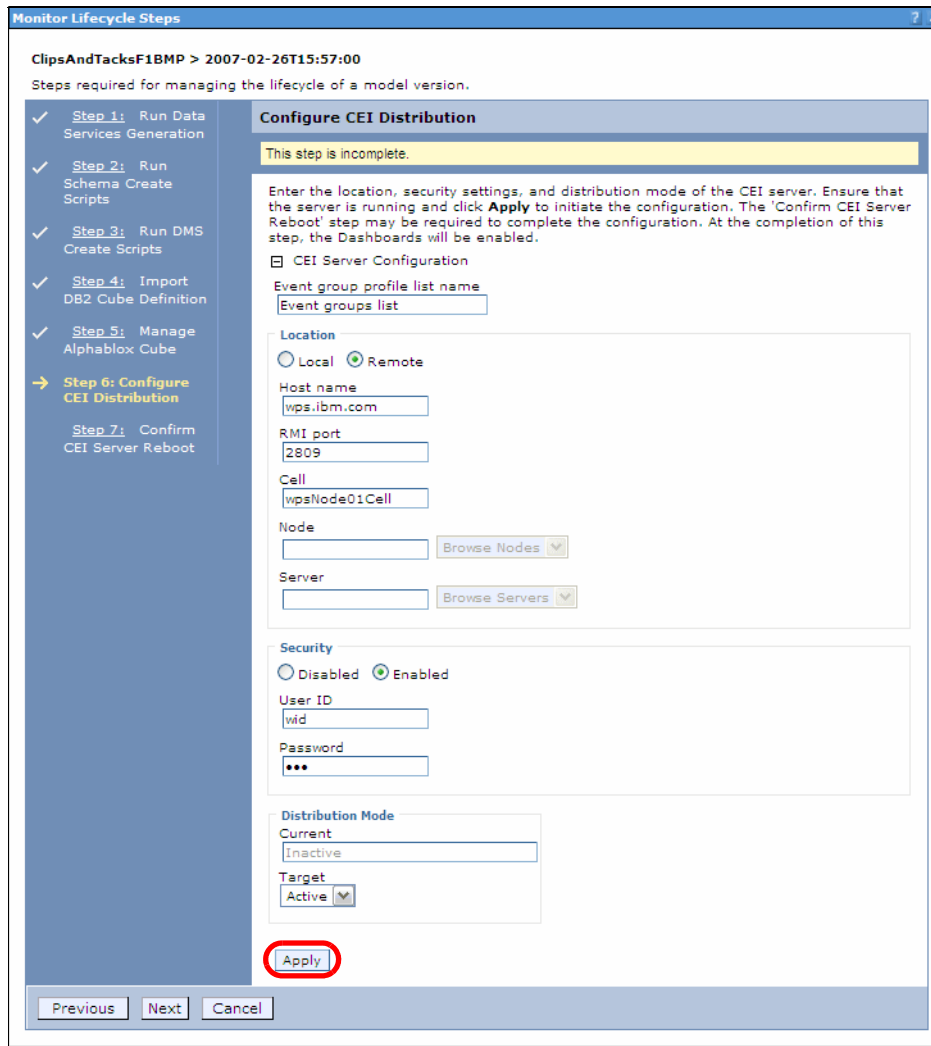


Figure 14-19 Monitor life cycle: Step 6

- ▶ Messages will be displayed to indicate that step 6 completed successfully (Figure 14-20).

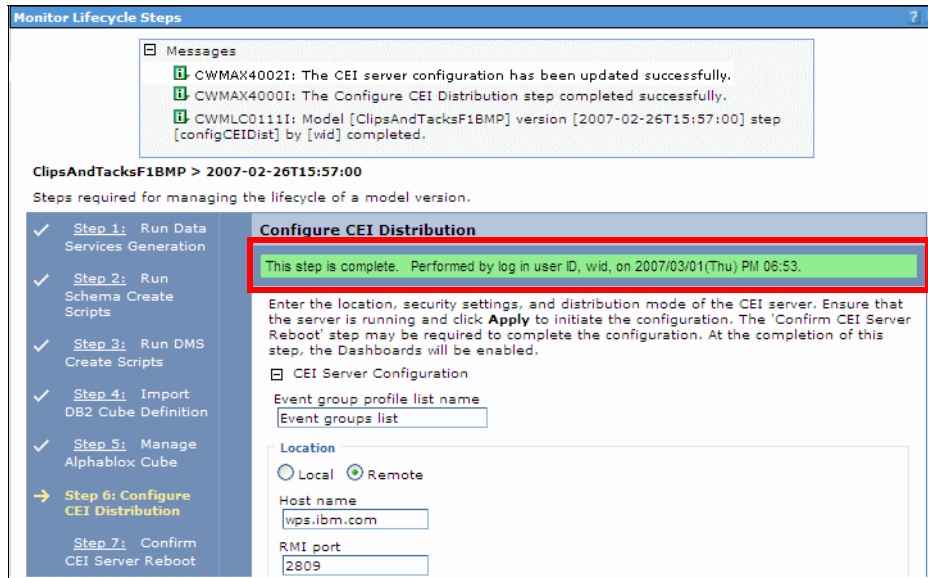


Figure 14-20 Monitor life cycle: Step 6 completed

Configure CEI in WebSphere Process Server

Prior to completing step 7 of the setup wizard, it is necessary to perform two configuration steps directly on the Process Server to complete the CEI configuration between the Process Server and the Monitor Server.

First we give the Process Server administrative user authority to write to the JMS queue that is shared between the Process Server and the Monitor Server. Then we perform configuration so that the application has authority to access the foreign destination on the Monitor Server:

- ▶ Launch the administrative console for the Process Server.
- ▶ Navigate to *Resources* → *JMS Providers* → *Default messaging*.
- ▶ Select *JMS queue connection factory* (Figure 14-21).

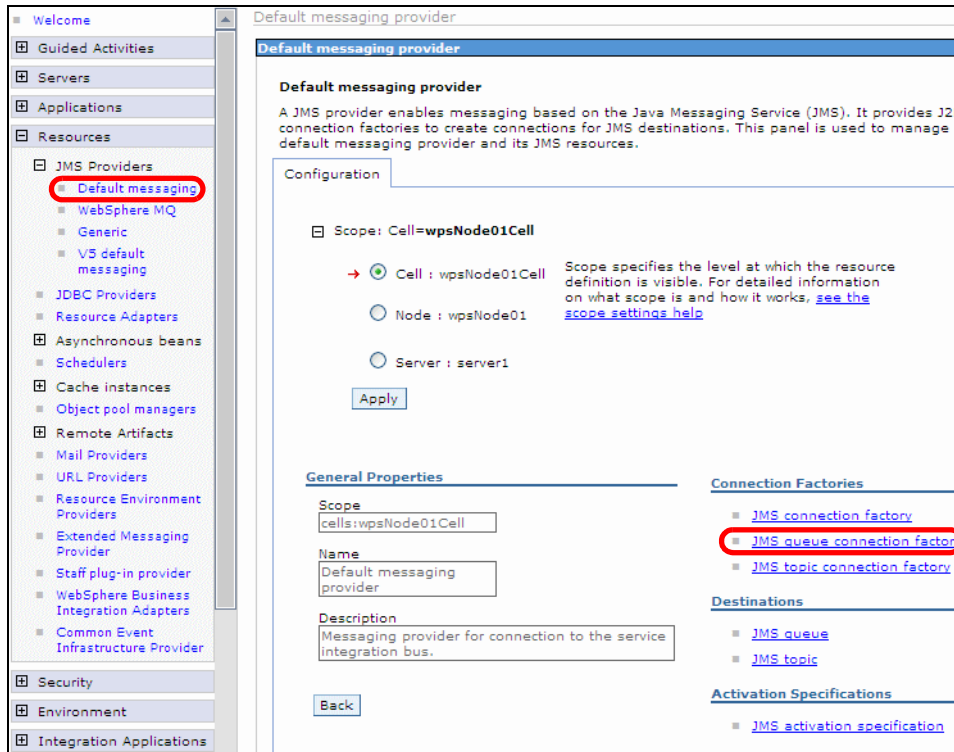


Figure 14-21 Configure JMS queue connection factory

- ▶ On the next panel (Figure 14-22), select:

mon_ClipsAndTacksF1BMP_1172505420_QF

This queue factory was automatically created as part of the CEI configuration performed by executing step 6 of the setup wizard.

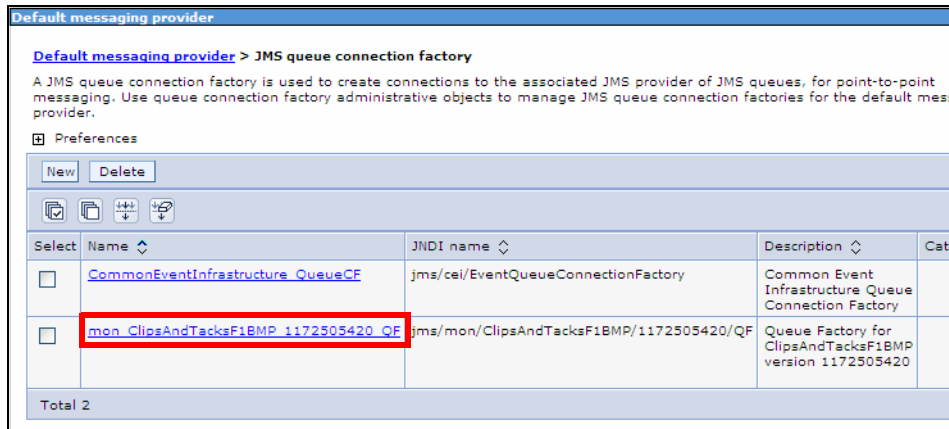


Figure 14-22 JMS queue connection factory for ClipsAndTacksF1BMP model

- ▶ On the following panel (Figure 14-23) scroll down to the bottom of the panel.

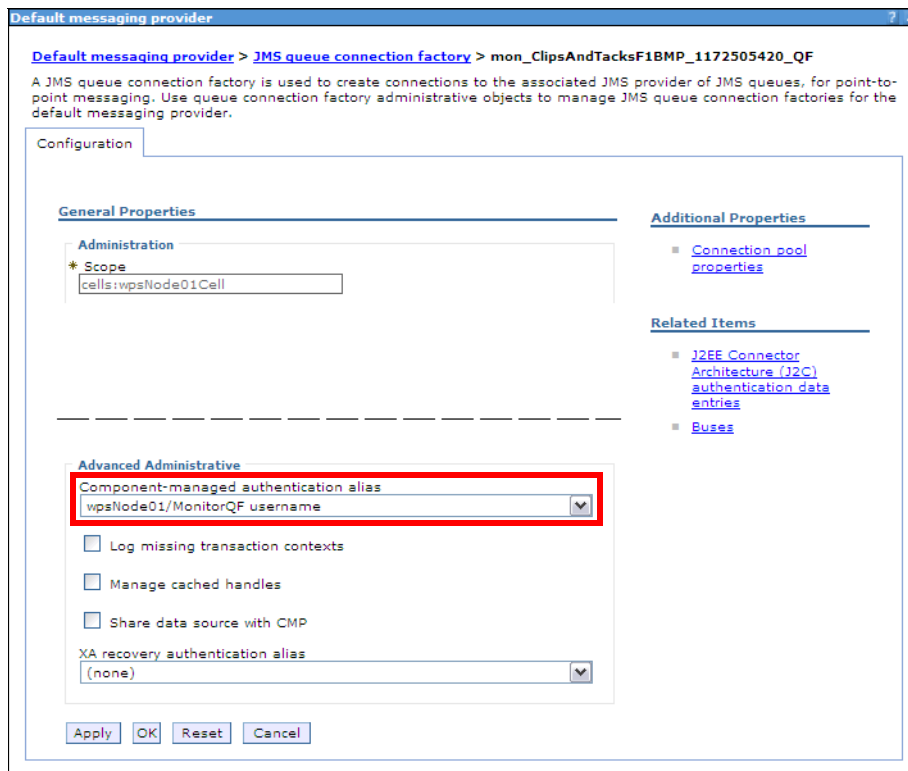


Figure 14-23 JMS queue connection factory: Authentication alias for writing to the queue

- ▶ In the *Advanced Administrative* section, select `wpsNode01/MonitorQF` username from the Component managed authentication alias section. Click *Apply* and save the changes.

This provides the Process Server user with authority to write to the JMS queue.

- ▶ Next we must run the following `wsadmin` command to configure the authentication between the application and the foreign destination created as part of the CEI configuration performed by executing step 6 of the setup wizard:

```
$AdminTask addUserToDestinationRole { -type foreignDestination
    -bus ${PROCESS_SERVER_BUS_NAME}
    -foreignBus ${MONITOR_SERVER_BUS_NAME}
    -destination ${Monitor_Bus_Queue_Destination}
    -role Sender
    -user ${USER_NAME} }
```

In our case, the `USER_NAME` is `wid`.

- ▶ First let us determine the values of the other three variables `${PROCESS_SERVER_BUS_NAME}`, `${MONITOR_SERVER_BUS_NAME}` and `${MONITOR_BUS_QUEUE_DESTINATION}`
 - In the WebSphere Process Server administrative console, navigate to *Service Integration* → *Buses* (Figure 14-24).
 - Select **MONITOR.wpsNode01Cell.Bus**. This is the `PROCESS_SERVER_BUS_NAME`.

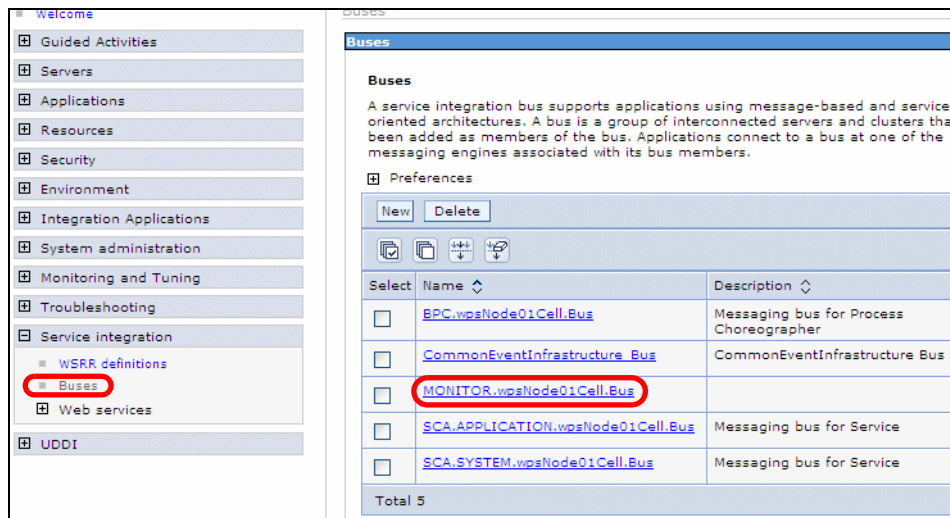


Figure 14-24 WebSphere Process Server Administrative Console: Buses

- In the next panel, select *Foreign buses* (Figure 14-25).

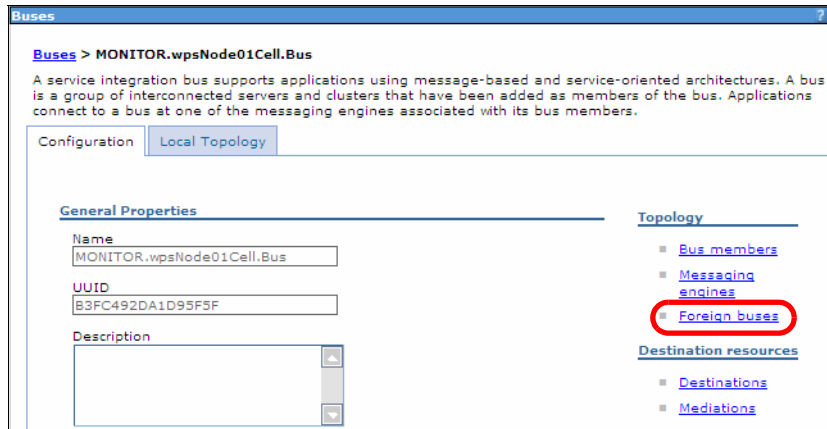


Figure 14-25 PROCESS_SERVER_BUS_NAME

- The foreign bus listed is the MONITOR_SERVER_BUS_NAME (Figure 14-26):
MONITOR.KLCHL2YN01C.Bus
- Select the Process Server bus, MONITOR.wpsNode01Cell.Bus to return to the previous panel (Figure 14-26).

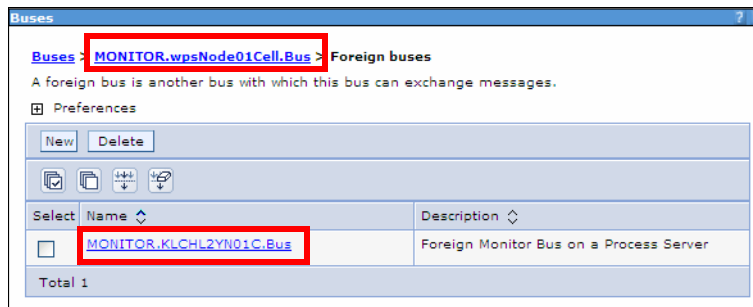


Figure 14-26 MONITOR_SERVER_BUS_NAME

- Select *Destinations* (Figure 14-27).

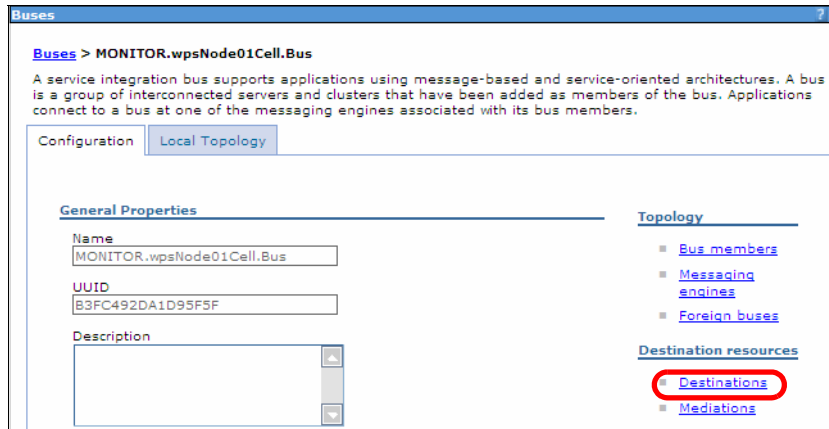


Figure 14-27 Local Monitor bus

- In the next panel (Figure 14-26), we see the foreign destination created for our monitor model as part of the CEI configuration step in the setup wizard:

mon_ClipsAndTacksF1BMP_1172505420_Q_Destination

This is the MONITOR_BUS_QUEUE_DESTINATION.

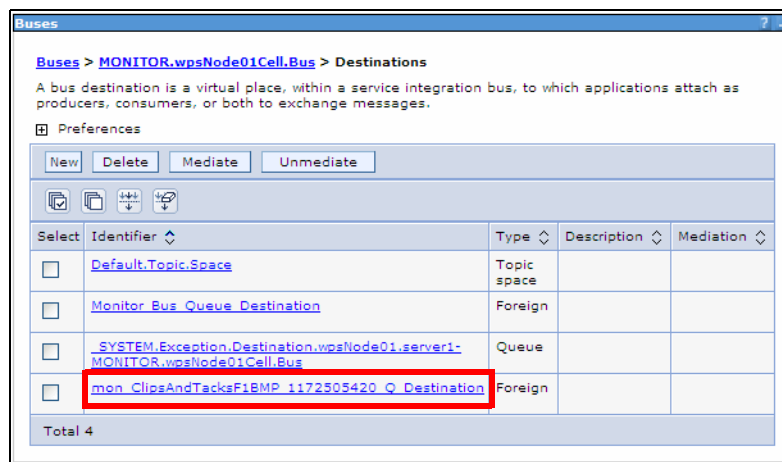


Figure 14-28 MONITOR_BUS_QUEUE_DESTINATION

We can now execute the wsadmin command with the correct values:

- ▶ On the WebSphere Process Server system, open a command prompt and navigate to the folder containing the wsadmin.bat script:

```
cd c:\IBM\WPS\profiles\ProcSrv01\bin
```


- ▶ In a command window, type wsadmin to open the wsadmin command prompt. You have to enter the user ID and password for the Process Server. Execute the command:

```
$AdminTask addUserToDestinationRole { -type foreignDestination
    -bus MONITOR.wpsNode01Cell.Bus
    -foreignBus MONITOR.KLCHL2YN01C.Bus
    -destination mon_ClipsAndTacksF1BMP_1172505420_Q_Destination
    -role Sender
    -user wid }
$AdminConfig save
```

The output should be as shown in Figure 14-29.

```
no accessible method "addUserToDestination" in class $Proxy1
  while executing
    "$AdminTask addUserToDestination < -type foreignDestination -bus MONITOR.wpsNode
01Cell.Bus -foreignBus MONITOR.KLCHL2YN01C.Bus -destination mon_Review0..."
wsadmin>$AdminTask addUserToDestinationRole < -type foreignDestination -bus MONI
TOR.wpsNode01Cell.Bus -foreignBus MONITOR.KLCHL2YN01C.Bus -destination mon_Revie
wOrderMonitorModel_1173736737_Q_Destination -role Sender -user wid>
<cells/wpsNode01Cell/buses/MONITOR.wpsNode01Cell.Bus!sib-authorisations.xml#SIBA
uthUser_1173230174132>
wsadmin>$AdminTask addUserToDestinationRole < -type foreignDestination -bus MONI
TOR.wpsNode01Cell.Bus -foreignBus MONITOR.KLCHL2YN01C.Bus -destination mon_Ship0
rderMonitorModel_1173736778_Q_Destination -role Sender -user wid>
<cells/wpsNode01Cell/buses/MONITOR.wpsNode01Cell.Bus!sib-authorisations.xml#SIBA
uthUser_1173230174132>
wsadmin>$AdminConfig save
```

Figure 14-29 Authentication with foreign destination: Output

Step 7: Confirm CEI Server Reboot

Step 7 involves rebooting the CEI server so that the changes associated with the previous step can be incorporated. The CEI server is the WebSphere Process Server where our business process application, ClipsAndTacksF1App, is running. Follow these steps:

- ▶ Restart the WebSphere Process Server where the business process is running.
- ▶ In the Monitor Server administrative console, go to step 7 of the setup wizard. Verify that the confirm button is enabled.
- ▶ Click *Confirm* (Figure 14-30).

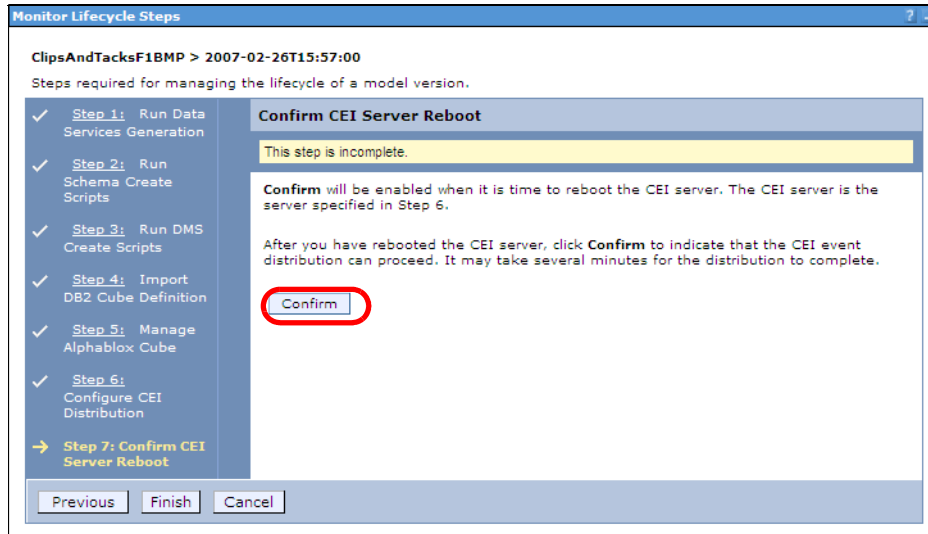


Figure 14-30 Monitor life cycle: Step 7

- Verify that the messages indicating successful completion of step 7 are displayed (Figure 14-31).

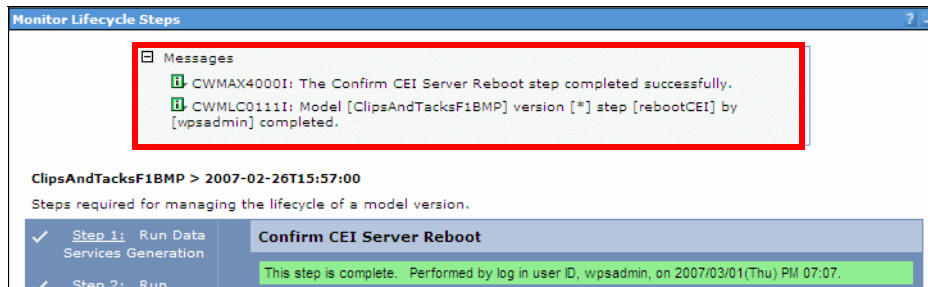



Figure 14-31 Monitor life cycle: Step 7 completed

- Click *Finish*.

Verify that the application is startable and started

Select *Applications* → *Monitor Models* and verify that the model is marked as both startable and running, indicated by the green icons (Figure 14-32).

If the model is not yet running, click the  icon to refresh the status. Be patient, this may take a while.

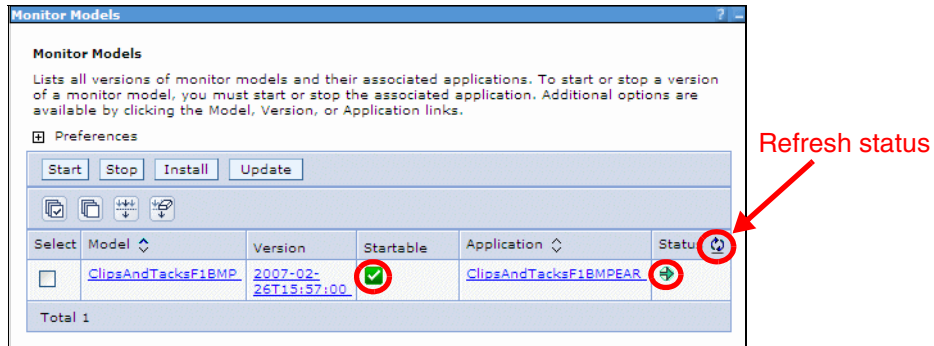


Figure 14-32 Monitor model: Successful deployment

The monitor model is now deployed and ready for use.

Configuring the Adaptive Action Manager

Next we configure the Adaptive Action Manager so that it will process the outbound situation events emitted by our monitor model and generate appropriate alerts for display in the Dashboard.

Connecting the Action Manager to LDAP

Follow these steps to define the connection to the LDAP Server:

- ▶ Open the administrative console for the Monitor Server.
- ▶ Navigate to *Applications* → *Monitor Action Manager* → *Configuration*.
- ▶ Take the defaults in the General tab, and select the *LDAP* tab. Update the fields to reference the LDAP server and specify uid for the LDAP alerts so that we can direct the alerts to specific dashboard users (Figure 14-33):
 - LDAP URL: ldap://localhost:389
 - LDAP user ID: cn=root
 - LDAP password: password
 - LDAP alert: uid
- ▶ Click *Apply*.

Note: You can verify the user ID and password by opening the Tivoli Directory Server directory configuration tool and selecting *Admin DN/password* on the left-hand side.

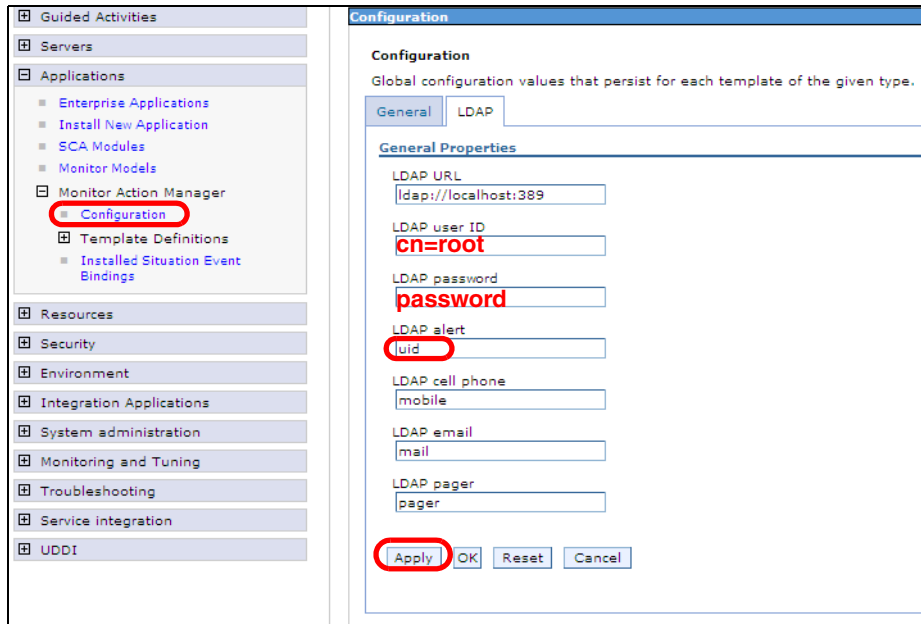


Figure 14-33 Monitor Action Manager configuration: LDAP configuration

Restart the Adaptive Action Manager application for the configuration changes to take effect. Navigate to *Applications* → *Enterprise Applications*. Stop the `IBM_WB_ACTIONMANAGER` application, then start it.

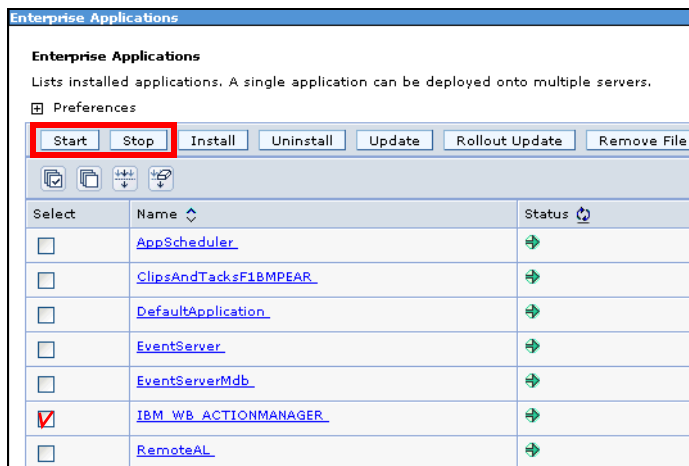


Figure 14-34 Stop and start the Adaptive Action Manager application

Defining the notification templates

We have to add a notification template to define the properties of the notification we want to send each time the order fulfillment time for a shipped order exceeds 3 days and 1 hour. Following this, we bind the notification to the situation event emitted by the monitor model when this situation arises.

To define the templates, follow these steps:

- ▶ Navigate to *Applications* → *Monitor Action Manager* → *Template Definitions* → *Notifications*.
- ▶ Click *New* (Figure 14-35).

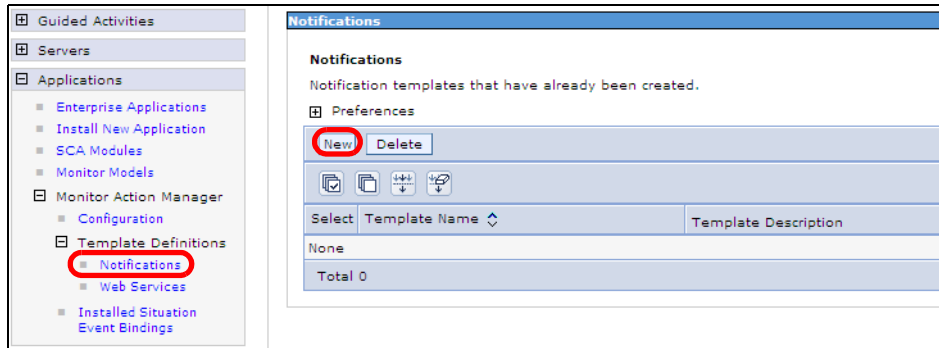


Figure 14-35 Monitor Action Manager configuration: Notification templates

- ▶ Define the notification template as follows (Figure 14-36):
 - Template name: OrderShipmentTimeExceeded
 - Action service type: Alert
 - To (LDAP query):
(&(uid=wpsadmin)(objectclass=top)(objectclass=inetOrgPerson))
 - Subject: Order exceeded target shipping time
 - Body: Order %Order.OrderNumber% exceeded target shipping time.
 - LDAP root: dc=ibm,dc=com

This choice of LDAP query means that only the wpsadmin user will receive the alert in the Dashboard alerts view. The variable %Order.OrderNumber% is replaced by the order number for the order that exceeded the target shipment time. This data was specified for inclusion in the extended data of the outbound event when the monitor model was created in “Shipped order fulfillment time is greater than 3 days and 1 hour” on page 362, and is extracted from the outbound situation event by the Adaptive Action Manager.

- ▶ Click *OK*.

The screenshot shows a web-based configuration interface for notifications. The title bar reads 'Notifications'. The main heading is 'Notifications > Notification Template Configuration'. Below this, it says 'Configuration properties for the notification template.' The configuration is for a template named 'OrderShipmentTimeExceeded'. Under 'General Properties', the 'Template name' is 'OrderShipmentTimeExceeded' and the 'Description' is empty. In the 'Action service type' section, the 'Alert' radio button is selected and circled in red. Other options are 'Cell phone', 'Email', and 'Pager'. A red box highlights the 'To (LDAP query)' field containing '(&(uid=wpsadmin))(objectclass=*)', the 'Subject' field containing 'Order exceeded target shipping', and the 'Body' field containing 'Order %Order.OrderNumber% exceeded target shipping time.'. Below these fields is the 'LDAP root' field containing 'dc=ibm,dc=com'. At the bottom, there are 'Apply', 'OK', 'Reset', and 'Cancel' buttons. The 'OK' button is circled in red.

Figure 14-36 Define OrderShipmentTimeExceeded Notification template

- ▶ Click *New* again to define a second notification template. This template is used to define the notification we want to send if the percentage of orders shipped falls below 85%.
- ▶ Complete the fields as before, but enter a meaningful template name, subject, and body. This time, the event data we want to display in the alert is the ship percentage, so we include %PercentageOfShippedOrders% in the alert body (Figure 14-37).

Notifications > Notification Template Configuration
Configuration properties for the notification template.

Notification Template Configuration

General Properties

Template name
ShipPercentageTooLow

Description

Action service type
 Alert
 Cell phone
 Email
 Pager

To (LDAP query)
(&(uid=wpsadmin))(objectclas

Subject
Too few orders shipped

Body
The shipment percentage is %PercentageOfShippedOrders% (below 85%)

LDAP root
dc=ibm,dc=com

Apply OK Reset Cancel

Note:
Better write:
(below 85 percent)
because % is used
for variables, and
does not show up in
the Dashboard.

Figure 14-37 Define ShipPercentageTooLow Notification template

As shown in Figure 14-38, we have now defined two notification templates.

Notifications
Notification templates that have already been created.

Preferences

New Delete

Select	Template Name	Template Description
<input type="checkbox"/>	OrderShipmentTimeExceeded	
<input type="checkbox"/>	ShipPercentageTooLow	

Total 2

Figure 14-38 Action Manager: Notification templates

Binding the notification templates to situation events

Next we must bind the situation events emitted by the monitor model to the notification templates defined above:

- ▶ Select *Applications* → *Monitor Action Manager* → *Installed Situation Event Bindings*.
- ▶ Click *New* (Figure 14-39).

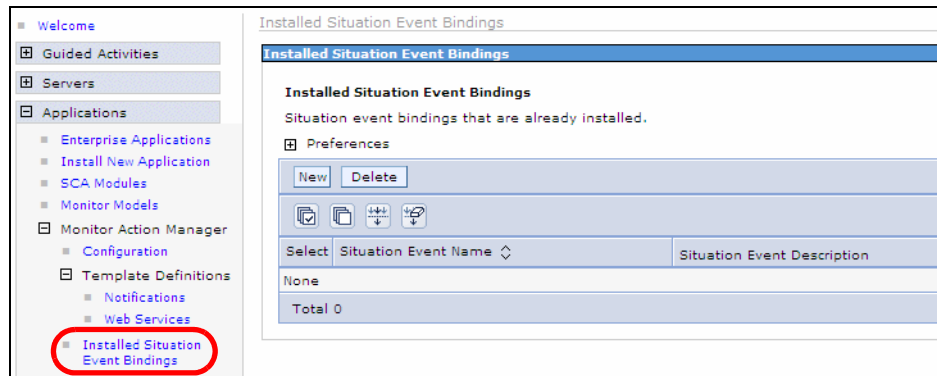


Figure 14-39 Adaptive Action Manager: Define situation event bindings

First define a situation event binding for the situation event that is emitted when the order fulfillment time for an order is greater than 3 days and 1 hour:

- ▶ Specify the situation event name. This must be identical to the value specified in the `BusinessSituationName` field in the outbound event when it was defined in the Monitor Model Editor (see Figure 12-57 on page 366).

Shipment is delayed

- ▶ Click *Apply* (Figure 14-40).

It is important that *Apply* is clicked prior to adding the binding to the notification template.

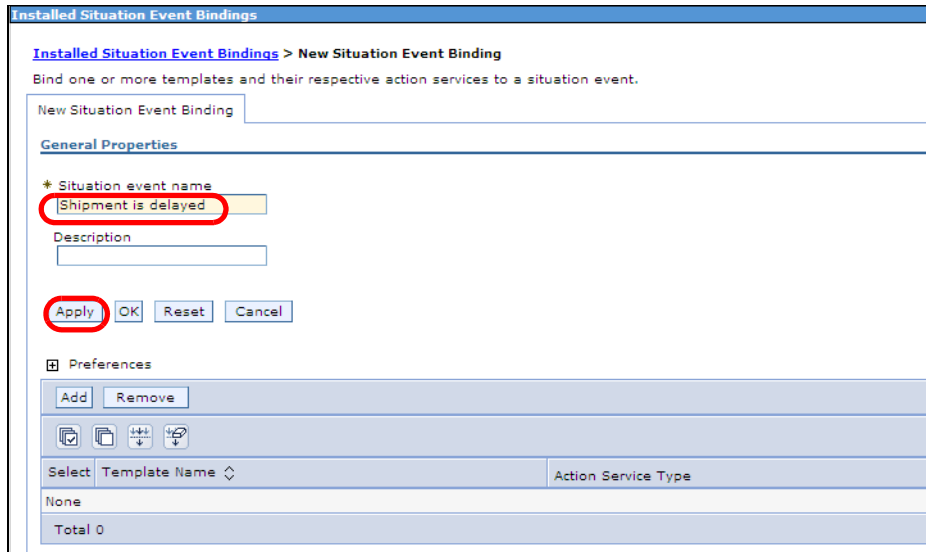


Figure 14-40 Binding between the alert and the notification

- ▶ Click *Add* to specify the template binding.
- ▶ In the prompt, select the `OrderShipmentTimeExceeded` template and click *OK* (Figure 14-41).

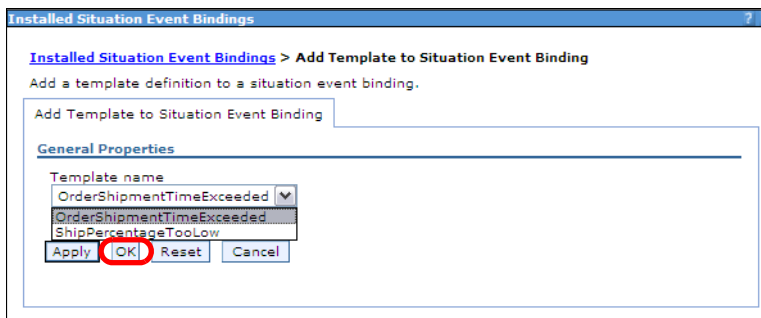


Figure 14-41 Adding template binding to situation event

- ▶ Back in the *New Situation Event Binding* panel, click *OK* (Figure 14-42).

Installed Situation Event Bindings

Installed Situation Event Bindings > New Situation Event Binding

Bind one or more templates and their respective action services to a situation event.

New Situation Event Binding

General Properties

* Situation event name
Shipment is delayed

Description

Apply **OK** Reset Cancel

Preferences

Add Remove

Select	Template Name	Action Service Type
<input type="checkbox"/>	OrderShipmentTimeExceeded	AlertHandler

Total 1

Figure 14-42 Completed situation event binding

- ▶ Click *New* to create a situation event binding for the other notification template.
- ▶ Again, ensure that the situation event name exactly matches the `BusinessSituationName` specified in the outbound event definition in the monitor model (Figure 12-60 on page 369). In this case:
 - Shipped Orders percentage too low
- ▶ Click *Apply*.
- ▶ Add the binding to the `ShipPercentageTooLow` template as before:
 - Click *Add*.
 - Select the `ShipPercentageTooLow` template.
 - Click *Apply*.
- ▶ Review the completed situation event binding and click *OK* (Figure 14-43).

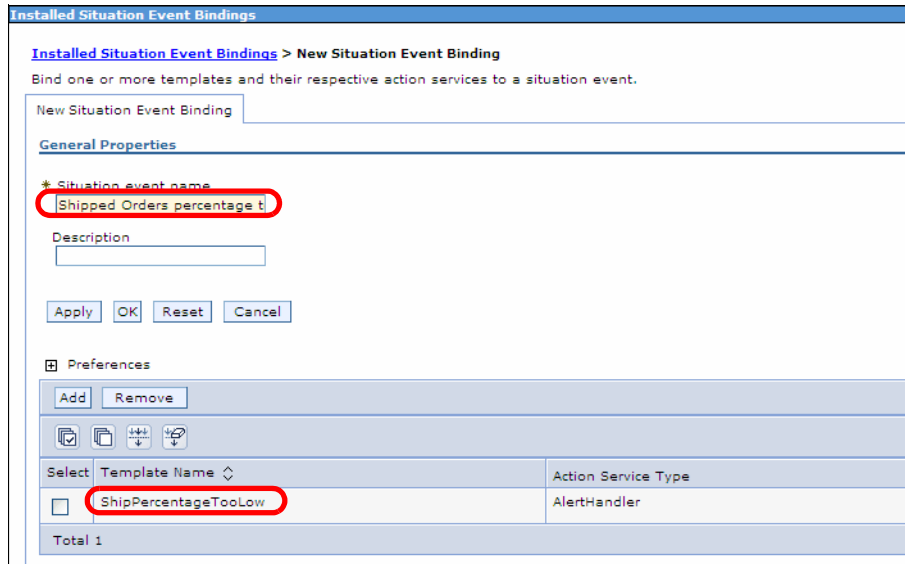


Figure 14-43 Ship Orders percentage too low situation event binding

Figure 14-44 shows the completed situation event bindings.

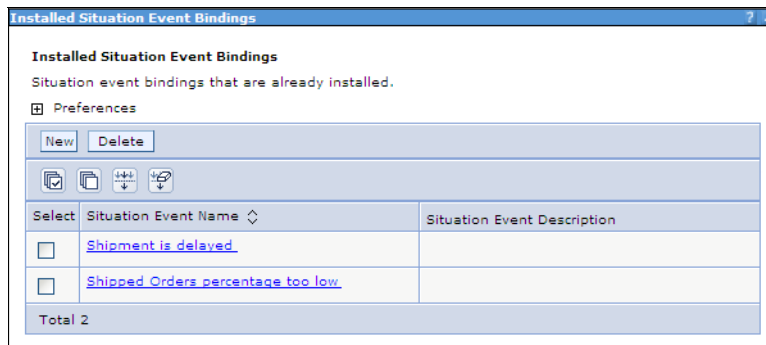


Figure 14-44 Action Manager: Situation event bindings

DB2 replication scripts

The DB2 replication scripts are included in the artifacts generated in step 1 of the monitor model deployment setup wizard. These replication scripts must be started in order to transfer data between the State tables in the MONITOR Database and the corresponding tables in the DATAMART database.

There are two scripts for each model:

- ▶ A capture script captures the necessary data from the MONITOR database.
- ▶ An apply script updates the DATAMART database.

To start the DB2 replicator scripts, follow these steps:

- ▶ Open a DB2 command prompt.
- ▶ Navigate to the directory containing the output from step 1 of the setup wizard. The StartCapture_#.bat script is in the subdirectory \DS_DMS_setup\State_to_Datamart\source\. In our case:

```
cd C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx\DS_DMS_setup\
State_to_Datamart\source\StartCapture_5.bat
```

- ▶ Execute the capture script:
StartCapture_5.bat
- ▶ A new window opens (Figure 14-45). This window must be left open or data replication will cease.

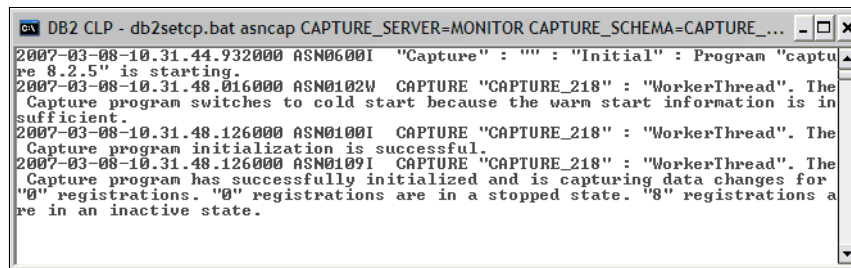


Figure 14-45 DB2 replication scripts: StartCapture

- ▶ Back in the original DB2 command window, change to the directory containing the StartApply_#.bat script:

```
cd C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx\DS_DMS_setup\
State_to_Datamart\target\StartApply_10.bat
```

- ▶ Execute the apply script (StartApply_10.bat) and leave the window open for replication to occur (Figure 14-46).

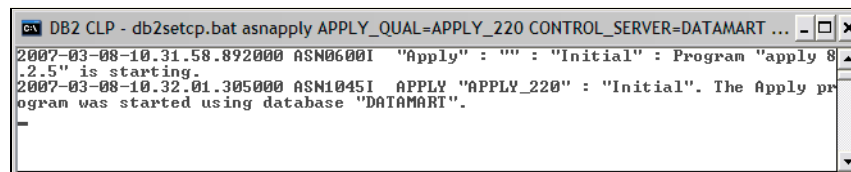


Figure 14-46 DB2 replication scripts: StartApply

Note: The StopCapture_#.bat and StopApply_#.bat scripts are provided to stop the replication daemons.

If you get an error when starting the capture script, see “Database replication problem” on page 484.

Monitor model undeployment

It is important that the monitor model is undeplayed correctly. Therefore, if you have to undeplay the monitor model at any point, follow these steps.

Stopping the monitor model

- ▶ Open the Monitor Models view in the administrative console (*Applications* → *Monitor Models*) and stop the model (Figure 14-47):
 - Select the check box next to the monitor model and click *Stop*.
 - Verify that the status of the model is updated to stopped.

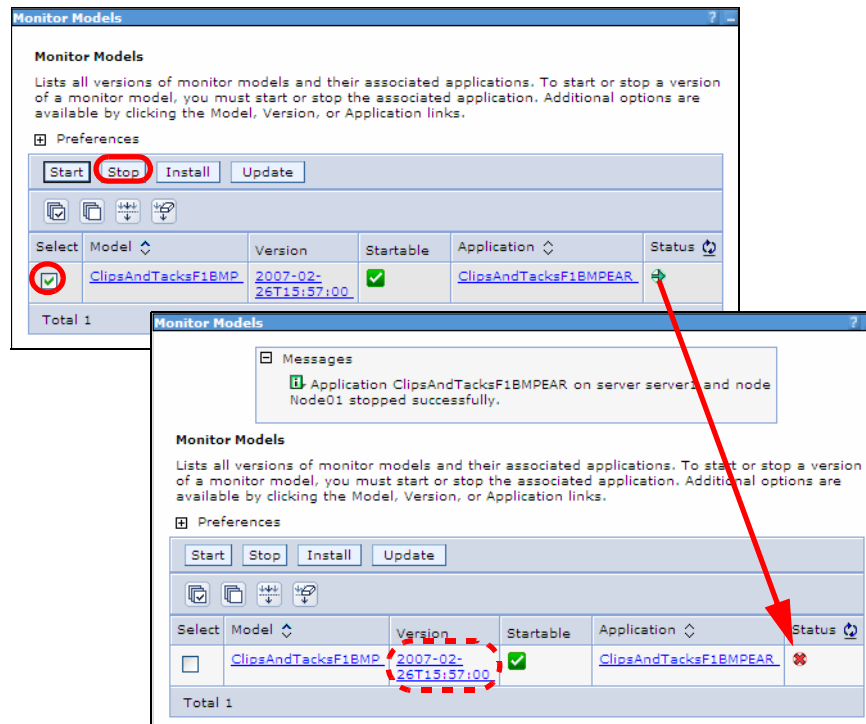


Figure 14-47 Monitor model undeployment: Stop the model

Removing the Alphablox cubes

Next remove the Alphablox cubes. Click the *Version* link (Figure 14-47), then click *Setup Wizard* (Figure 14-48).

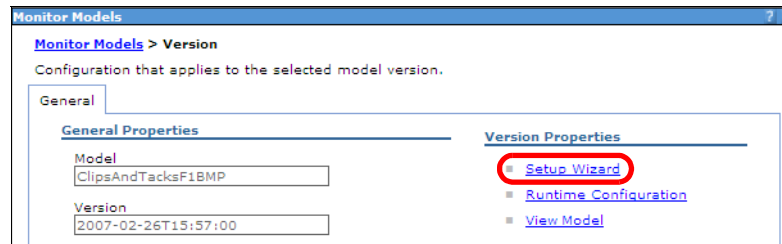


Figure 14-48 Monitor model undeployment: Setup Wizard

- ▶ **Ensure that the Monitor Dashboard Portal Server is running.**
- ▶ In the setup wizard, select *Step 5* (Figure 14-17 on page 430).
- ▶ In step 5 of the setup wizard, re-enter the password for the Administrator on the WebSphere Application Server that the Dashboard Portal is using (wpsbind). Then click *Remove* (Figure 14-49). A message is displayed indicating that the Alphablox cube(s) were successfully removed.

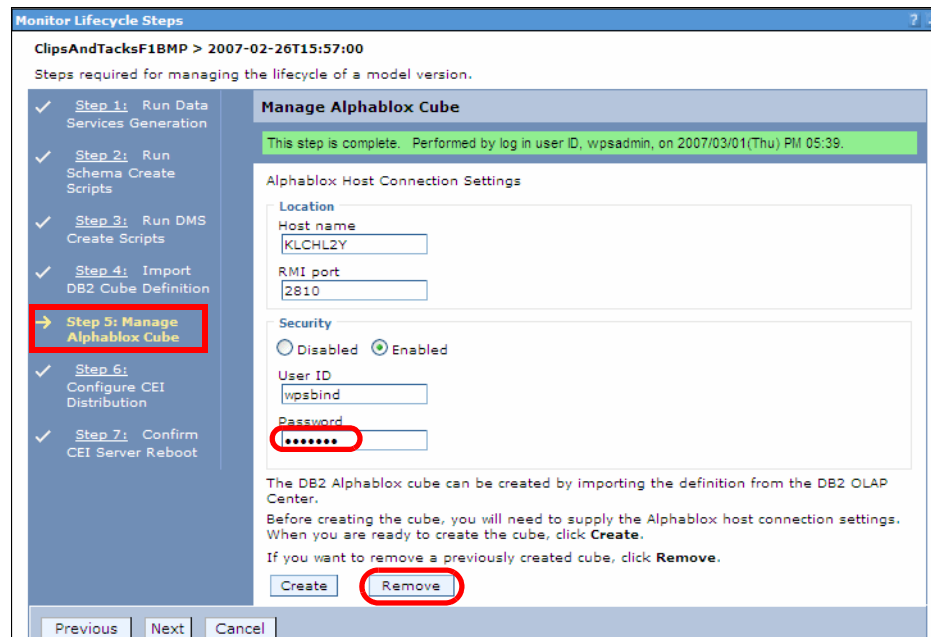


Figure 14-49 Monitor model undeployment: Removing the Alphablox cube(s)

Removing the DB2 Cube Views configuration

Removal of the DB2 Cube Views configuration (created in Step 4: Import DB2 Cube Definition of the setup wizard) is a command line operation performed outside of the WebSphere administrative console. Follow these steps:

- ▶ Open a DB2 command window.
- ▶ Navigate to the directory containing the output produced by the Data Services Generation (“Step 1: Run Data Services Generation” on page 420) when the model was originally deployed.

```
cd C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx
```

- ▶ Execute the following command:

```
db2mdapiclient -d datamart -u db2admin -p password -i drop_model_cv.xml  
-o drop_response.xml
```

Edit the command as necessary if your DATAMART database has a different name or your DB2 administrator logon credentials are different.

- ▶ The following result should be returned:

```
====> SQLCODE(0) SQLSTATE( ) :  
SQL08026
```

Removing the DMS configuration

Next we remove the data movement services configuration carried out in Step 3: Run DMS Create Scripts of the setup wizard. Removal of the DMS configuration is also a command line operation performed outside of the WebSphere Administrative Console. Follow these steps:

- ▶ Ensure that the replication scripts (StartCapture_#.bat and StartApply_#.bat) have been stopped.

- To stop the capture script execute the StopCapture_x command:

```
C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx\DS_DMS_setup\  
State_to_Datamart\source\StopCapture_6.bat
```

- To stop the apply script:

```
C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx\DS_DMS_setup\  
State_to_Datamart\target\StopApply_11.bat
```

- ▶ Unzip the DMSCleanup.zip file in:

```
C:\IBM\WebSphere\Monitor\tools\DataMovementServices\DMSCleanup.zip
```

- ▶ Run the following command:

```
dmscleanup -model ClipsAndTacksF1BMP
```

You are prompted for database names, user ID, and password.

Note: This command removes the DMS configuration for all versions of the model. In our case we only have one version of the ClipsAndTacksF1BMP model installed.

Removing the model schema

Removal of the monitor model schema and all associated data is also performed as a command line operation. We drop the Monitor and Data mart databases by running the `stateDrop.ddl` and `datamartDrop.ddl` scripts which were created in Step 1: Run Data Services Generation of the setup wizard.

- ▶ Navigate to the directory containing the `stateDrop.ddl` and `datamartDrop.ddl` scripts:

```
C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx
```

- ▶ Open both scripts and edit them to uncomment all the commands. When the scripts are created the commands are commented out to avoid the user dropping the databases by accident.
- ▶ Open a DB2 command window.
- ▶ Navigate to the directory containing the output from the data services generation:

```
cd C:\ClipsAndTacksF1\schemagen\2007_xx_xx_xxxx_xx
```

- ▶ Execute the following commands:

```
db2 connect to monitor  
db2 +c -tvf stateDrop.ddl
```

- ▶ Open a second DB2 command window and execute the following commands:

```
db2 connect to datamart  
db2 +c -tvf datamartDrop.ddl
```

- ▶ Following successful completion of both commands, return to the first window and enter these commands:

```
db2 commit  
db2 disconnect monitor
```

- ▶ Similarly, in the second window, enter these commands:

```
db2 commit  
db2 disconnect datamart
```


Uninstalling the monitor model version application

Finally we uninstall the application for the monitor model. Follow these steps:

- ▶ Back in the monitor models section of the monitor server administrative console, take a note of the application name associated with the model (Figure 14-50).

Note that as a result of the steps above, the monitor model is no longer startable.

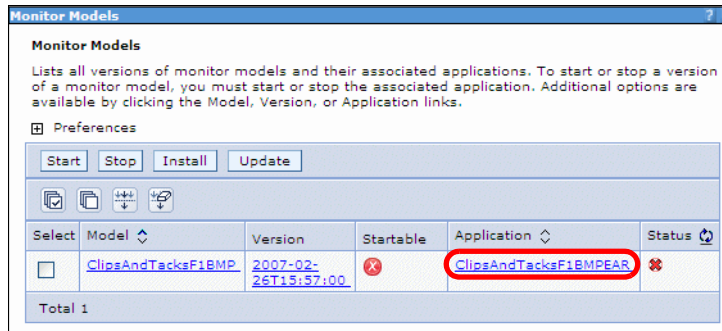


Figure 14-50 Monitor model undeployment: Application name

- ▶ Select *Enterprise Applications*.
- ▶ Select the check box next to the monitor model application `ClipsAndTacksF1BMPEAR` and click *Uninstall* (Figure 14-51).

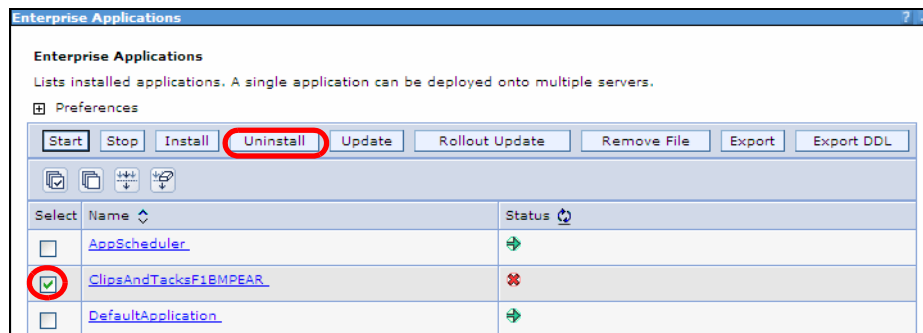


Figure 14-51 Monitor model undeployment: Uninstall the application

- ▶ In the Uninstall Application confirmation window, click *OK*.
- ▶ Click *Save* to save the changes to the master configuration.
- ▶ Select *Applications* → *Monitor Models* to return to the list of monitor models. The monitor model should no longer be displayed.

Troubleshooting

If a monitor model version application is uninstalled prior to dropping the databases, then it is not possible to completely remove the version from the Monitor databases. In this situation, the monitor model is still displayed when the monitor model application is uninstalled (Figure 14-52).

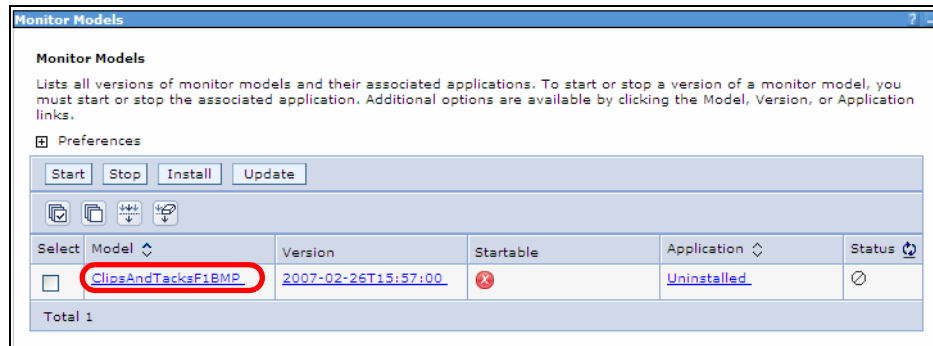


Figure 14-52 Monitor model undeployment: Monitor model requires purging

In this situation, if all versions of the monitor model have been uninstalled, the model can be purged:

- ▶ Click the model name, ClipsAndTackF1BMP.
- ▶ On the next page, click *Purge* (Figure 14-53)

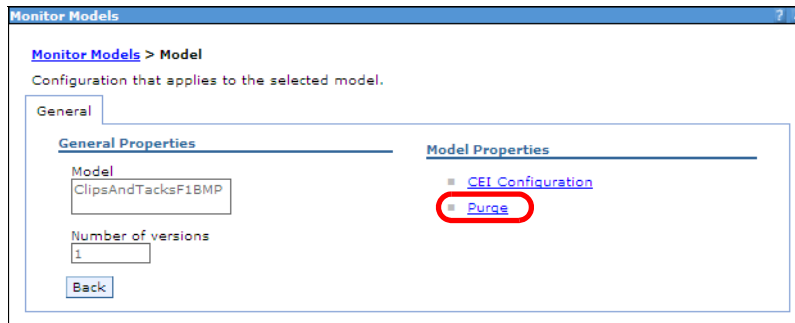


Figure 14-53 Monitor model undeployment: Purge command

- ▶ On the following panel, click *Purge* to confirm (Figure 14-54).

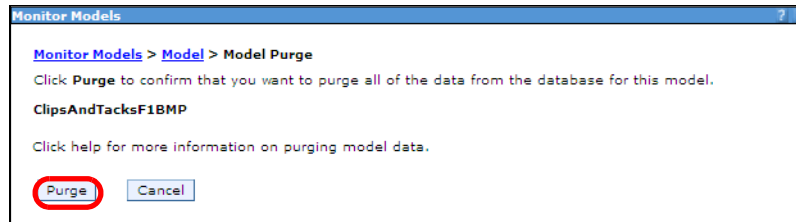


Figure 14-54 Monitor model undeployment: Confirm purge command

- ▶ The Purge page will be refreshed to display the confirmation messages (Figure 14-55).

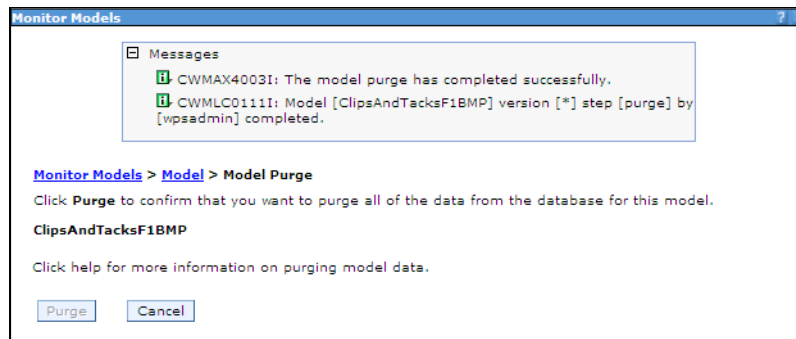


Figure 14-55 Monitor model undeployment: Confirmation of purge

- ▶ Return to the monitor models page and verify that the monitor model is no longer listed.

What next?

You can now run the business process application and accumulate instance data, or you can configure the dashboard first.

In practice, you would run the application and in parallel configure the dashboard. This has the advantage that you can immediately see the dashboard with actual data.

We perform some basic dashboard configuration first, then run the application, and then measure the application is the dashboard. We can always reconfigure the dashboard to suit our needs.

WebSphere Business Monitor Dashboard

When modelling the business process, the business analyst specified the monitoring requirements of the business using WebSphere Business Modeler (see Chapter 9, “Defining KPIs and measures” on page 187). The business measures for the ClipsAndTacks model were then discussed in detail in Chapter 12, “Developing and testing the business measures with the Monitor Toolkit” on page 327.

After modelling and deploying the process and creating and deploying the monitoring model, the business measures gathered from the runtime engine are displayed to the user using a set of predefined dashboards provided by WebSphere Business Monitor. Refer to “Monitor Dashboard” on page 57 for a discussion about the different dashboards provided by the product.

Based on the business measures requirements for the ClipsAndTacks application, the following dashboard views are used:

- ▶ Alerts view
- ▶ Dimensional view
- ▶ Export values view
- ▶ Gauges view
- ▶ Instances view
- ▶ Key performance indicator view

The shipped dashboard views require configuration to display the required data. For example, we have to specify the monitoring model to be used as the source of the business measures data, and specify which business measures should be displayed in the dashboard. During the process of configuring the dashboards, it is necessary to perform some administrative activities on WebSphere Portal.

There are three main steps involved in configuring the dashboard so that it presents the business user with the required information:

- ▶ Make a copy of each of the WebSphere Business Monitor views to be used.
- ▶ Create the portal page and add the portlets (views) to it.
- ▶ Configure the dashboard views for the monitor model.

Copying the WebSphere Business Monitor portlets

To copy the dashboard view, perform these steps:

- ▶ Login to WebSphere Portal administrative console (Figure 14-56):
`http://localhost:9080/wps/portal` (your port may be different)

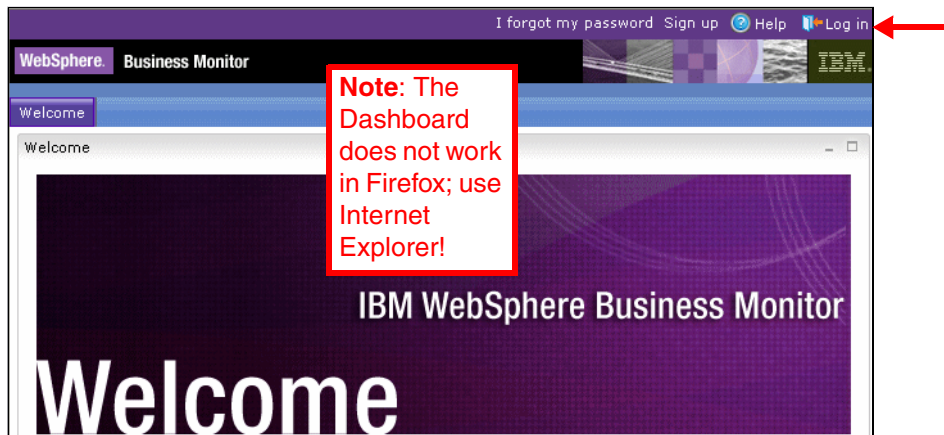


Figure 14-56 WebSphere Portal login screen

- ▶ Click *Log in* and provide the Portal administrator user name and password (in our case `wpsadmin/wpsadmin`).
- ▶ To verify that you logged in as an administrator, check the top right pane of the page. If it contains the *Administration* link, then you are logged as an administrator (Figure 14-57).



Figure 14-57 Administration link

- ▶ Click *Administration*.
- ▶ On the next page, select *Portlet management* → *Portlets* in the left pane (Figure 14-58).

There are 11 pages of portlets listed. The Monitor portlets are typically at the end of the list.

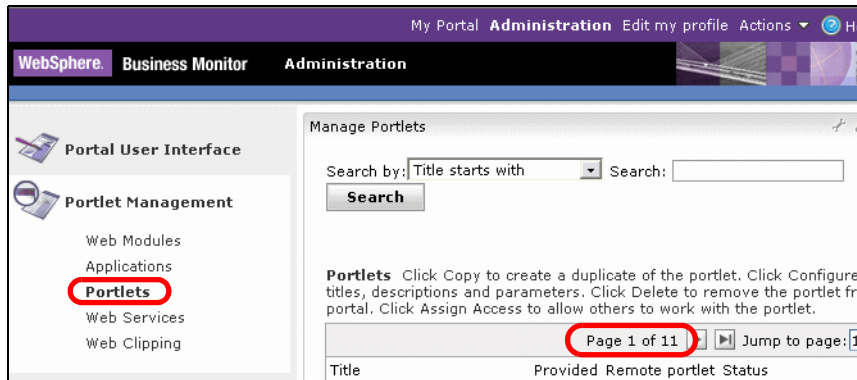



Figure 14-58 Portlets listing

To copy the Monitor portlets, we use the search facility:

- ▶ In the search field, type Alerts and click *Search*.
- ▶ Select the copy portlet icon  (Figure 14-59).

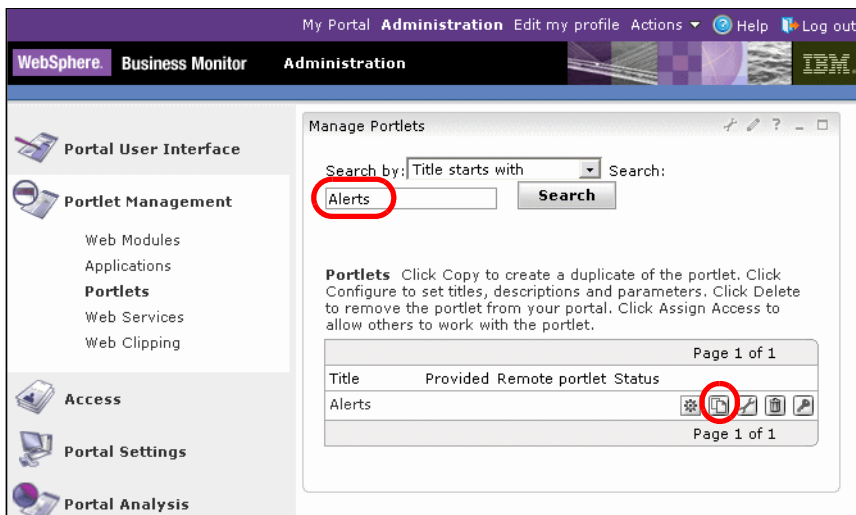


Figure 14-59 Portlet search result

- ▶ On the next page, enter a name for the new copy of the alerts view (Figure 14-60):

Clips And Tacks Alerts

This name will be displayed on the dashboards, therefore we recommend that you use the model name followed by the view name.

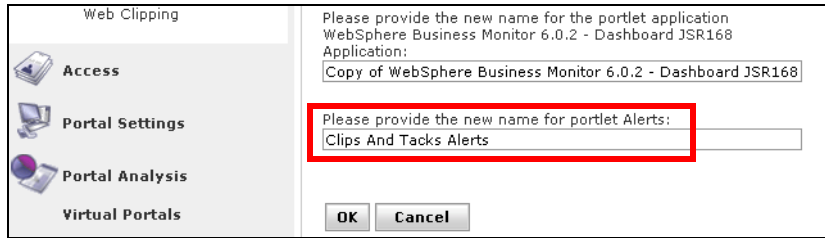


Figure 14-60 Selecting a name for the new copy of the Alerts portlet

- ▶ Repeat these steps to create a copy of each of the views we want to use, with the following naming:
 - Dimensional view: Clips and Tacks Order Location
 - Export value view: Clips and Tacks Export actual values
 - Gauge view: Clips and Tacks Gauge view
 - Instances view: Clips and Tacks Instances
 - Key performance indicator view: Clips and Tacks KPI view
- ▶ To verify, search for portlets with Clips and Tacks in the name (Figure 14-61).

Page 1 of 1	
Title	Provided Remote portlet Status
Clips and Tacks KPI view	
Clips and Tacks Gauge view	
Clips and Tacks Export actual values	
Clips And Tacks Alerts	
Clips and Tacks Instances	
Clips and Tacks Order Location	
Page 1 of 1	

Figure 14-61 List of ClipsAndTacks portlets copied

Creating a portal page for ClipsAndTacks

We have to assemble the portlets into a portal page.

- ▶ Login to the portal as an administrator.
- ▶ Select *Portal User Interface* → *Manage Pages* → *My Portal* (Figure 14-62).

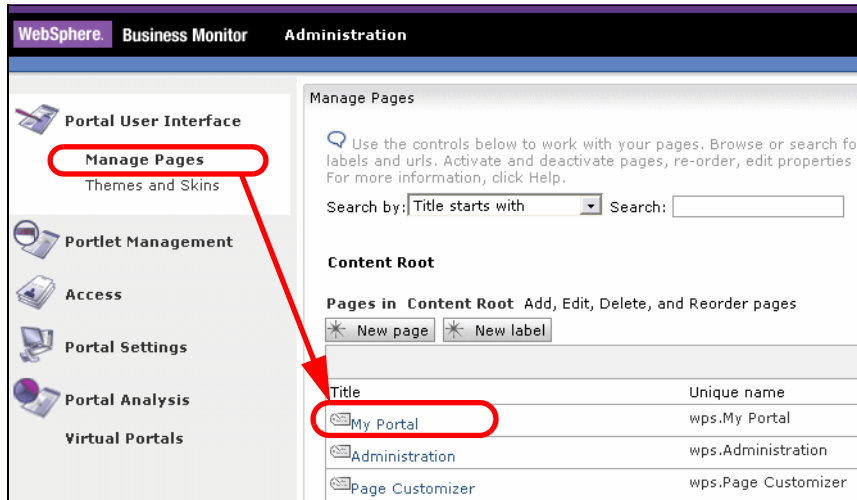


Figure 14-62 Accessing My Portal

Create the ClipsAndTacks portal page

To add a ClipsAndTacks portal page, perform these steps:

- ▶ Click *New Page*.
- ▶ In the Title Field, type Clips And Tacks and click *OK*.
- ▶ The new page should now be listed (Figure 14-63).

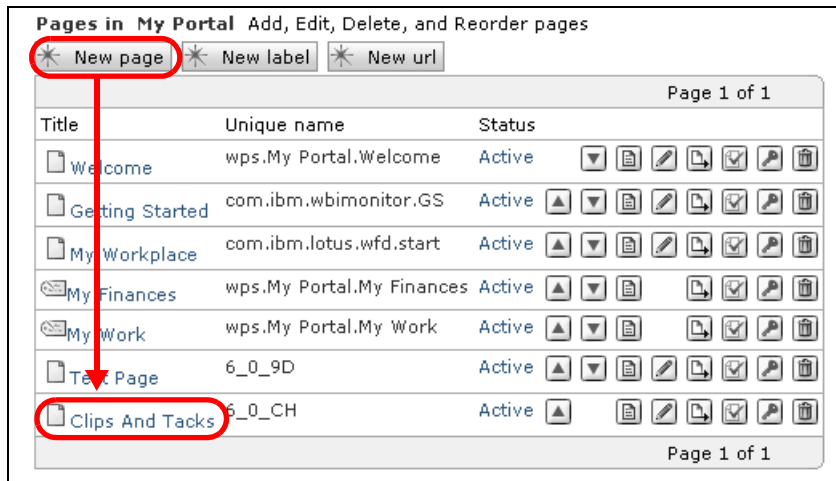


Figure 14-63 Clips And Tacks page

Create pages in the ClipsAndTacks portal

We could add portlets directly to the portal page, but it is advisable to create subpages with one or two portlets on each page:

- ▶ Select the *Clips And Tacks* portal page, then click *New page* and type *Instances* as title. Click *OK* and the *Instances* page is added (Figure 14-64).

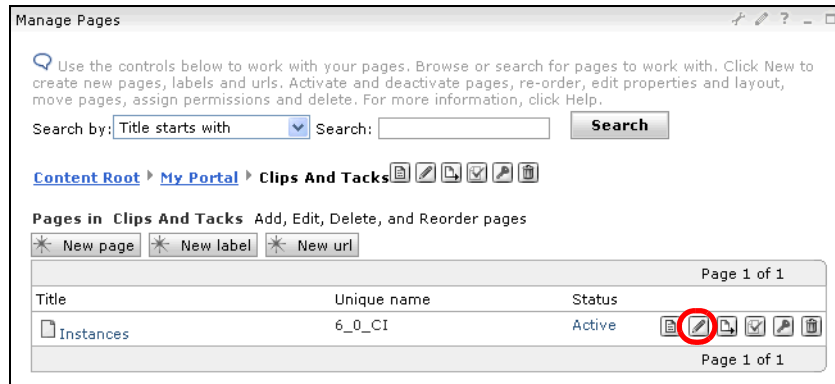



Figure 14-64 Creating portal pages for ClipsAndTacks

- ▶ Click the *Edit Page Layout* icon .
- ▶ The Edit page layout page opens. Click *Add Portlets*.
- ▶ The list of portlets appears. Search for *clips* to list our portlets.
- ▶ First we will add the instances view. Select the *Clips and Tacks Instances* portlet and click *OK* (Figure 14-65).

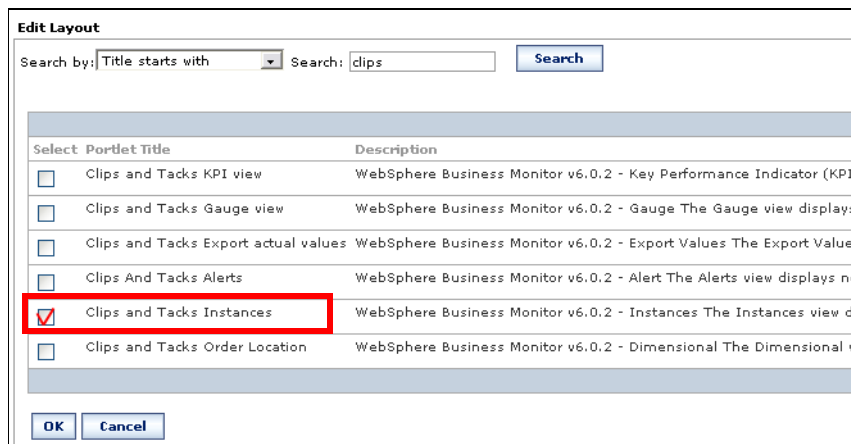


Figure 14-65 Selecting portlet to be added to a page

- ▶ The selected portlet is added. Notice that you can select different page layouts. Click *Done* (Figure 14-66).

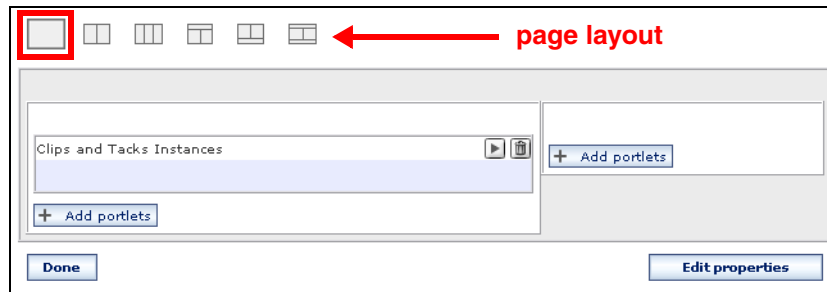


Figure 14-66 Portal page with one portlet

- ▶ On the top of the page, click *My Portal*.
- ▶ Select the *Clips And Tacks* page, then click *Instances* to verify that the instances view has been added (Figure 14-67).

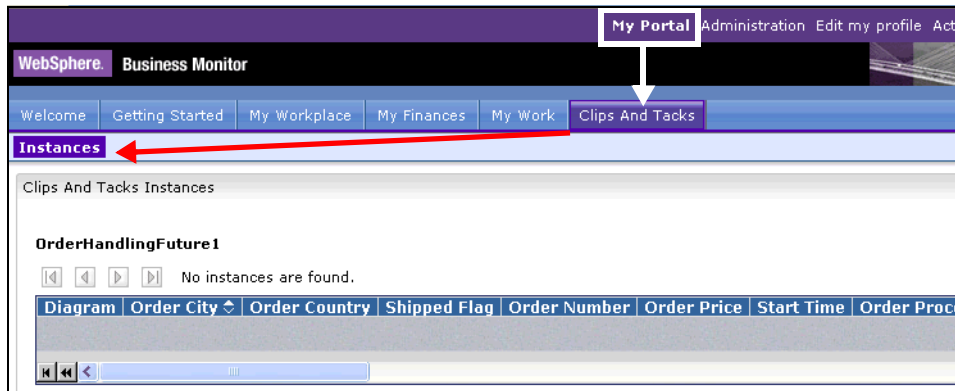


Figure 14-67 Clips And Tacks Instances page

- ▶ Return to the Clips And Tacks page in the Administration section:
 - Select the *Administration* link at the top of the page.
 - Navigate to *Portal User Interface* → *Manage Pages* → *My Portal* → *Clips And Tacks*.
- ▶ Add the remaining views to the dashboard (Figure 14-68):
 - Click *New page* and enter a title. Click *OK*.
 - Click the *Edit Page Layout* icon.
 - Click *Add Portlets*.
 - The list of portlets appears. Search for *clips* to list the portlets.
 - Select the matching portlet and click *OK*.

– Click *Done*.









































Page 1 of 1		
Title	Unique name	Status
Instances	6_0_CI	Active      
KPI	6_0_CJ	Active       
Gauge	6_0_CK	Active       
Location	6_0_CL	Active       
Alert	6_0_CM	Active       
Export	6_0_CN	Active      
Page 1 of 1		

Figure 14-68 ClipsAndTacks dashboard pages


Note that you could select multiple portlets in one page, but for now we create pages with one portlet.

Configuring the dashboard views

The dashboard pages contain all the required views. Now we have to configure each portlet.

Instances view configuration

From the Clips And Tacks dashboard page, we configure each page:

- ▶ Select *My Portal* → *Clips And Tacks*, then select *Instances*.
- ▶ At the right of the portlet there is a set of icons. Click the *configure* icon  (Figure 14-69).

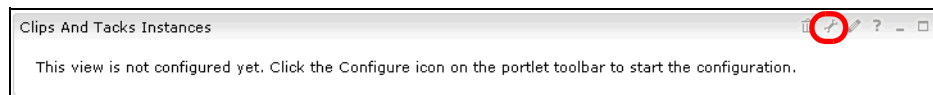


Figure 14-69 Configure Instances view

- ▶ Select the model, monitoring context, and the metrics required to be displayed, then click *Next* (Figure 14-70).

Clips And Tacks Instances

Select a monitor model, a monitoring context, and one or more metrics to be shown in the View mode.

Monitor model:
ClipsAndTacksF1BMP

Monitoring context:
OrderHandlingFuture1

Metrics:

Select All

Process Instance ID Order City Order Country Shipped Flag

Order Number Order_Count Order Price Shipped Order %

Approve Without Review % Start Time Order Process Time

Metrics for administration:

Process Instance ID Order City Order Country Shipped Flag

Order Number Order_Count Order Price Shipped Order %

Approve Without Review % Start Time Order Process Time

Back Next Finish Cancel

Figure 14-70 Selecting metrics to be displayed in the instances view

- ▶ On the next page, you specify sorting and the number of instances to be displayed per page (Figure 14-71).

Metric Name	Has Index	Sortable	Default Sorting	Sort Order	Monetary Metric	Currency
Order City	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="radio"/>	Ascending	<input type="checkbox"/>	
Order Country	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Ascending	<input type="checkbox"/>	
Shipped Flag	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Ascending	<input type="checkbox"/>	
Order Number	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Ascending	<input type="checkbox"/>	
Order Price	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Ascending	<input type="checkbox"/>	
Start Time	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Ascending	<input type="checkbox"/>	
Order Process Time	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Ascending	<input type="checkbox"/>	

Number of rows to display: 10

Back Next Finish Cancel

Figure 14-71 Configuration of the Instances view

- ▶ On the next page you specify whether to display all instances or only the active instances. You can also filter the list of instances based on values of the metrics (Figure 14-72).

Clips And Tacks Instances

Optionally, use this screen to limit the displayed instances. You can create a filter by typing or pasting the expression or both. Click Validate to make sure that the expression is syntactically correct.

Show: **All instances** all or active instances

Filter by metric:

Shipped Flag Matching case sensitive

Shipped Flag = "SHIPPED" optional

()

Description:

Figure 14-72 Instances view filter

- ▶ Prior to confirming the configuration, the view is displayed so that the configuration can be verified. Click *Finish*.

Key performance indicator view configuration

The configuration that can be performed with the KPI view is to modify the target and the ranges of the view.

- ▶ Click *Configure* (icon) for the KPI view.
- ▶ The list of KPIs is displayed. Click the *Modify* icon to configure the KPI (Figure 14-73).

Clips and Tacks KPI view

KPI Name	Monitor Model	Modify
Average Shipped Order Fullfillment Time	ClipsAndTacksF1BMP	<input type="button" value="✎"/>
Shipped Order %	ClipsAndTacksF1BMP	<input type="button" value="✎"/>

Figure 14-73 Key performance indicators available for configuration

- ▶ You can modify the target and edit the values of the ranges, but you cannot add or delete a range (Figure 14-74). Creating or deleting a range must be performed in edit mode.

Clips And Tacks KPI view

KPI Name:
Shipped Order %

Description:
Details

Target and Ranges

Target
90

Range Type
Actual Value

Ranges

Range Name	Start Value	End Value
Range1	0	<85
Range2	85	<90
Range3	90	<95
Range4	95	<100


OK Cancel

Figure 14-74 Modifying KPI target and range values

- ▶ Click *OK* or *Cancel*, then click *Exit*.

Personalizing a KPI

Personalization in the KPI view means configuring the ranges, including the ability to add or delete a range, and selection of the colors.

To personalize the KPI view, click Edit  (Figure 14-69 on page 465):

- ▶ Select *Show in View* for both KPIs.
- ▶ Click *Personalize* for Average Shipped Order Fulfillment Time (Figure 14-75).

Clips And Tacks KPI view



KPI Name	Monitor Model	Show in View	Personalize
Average Shipped Order Fulfillment Time	ClipsAndTacksF1BMP	<input checked="" type="checkbox"/>	
Shipped Order %	ClipsAndTacksF1BMP	<input checked="" type="checkbox"/>	

Figure 14-75 KPI personalization

- ▶ The target (3 days) and the existing four ranges are displayed with their default colors (Figure 14-76).

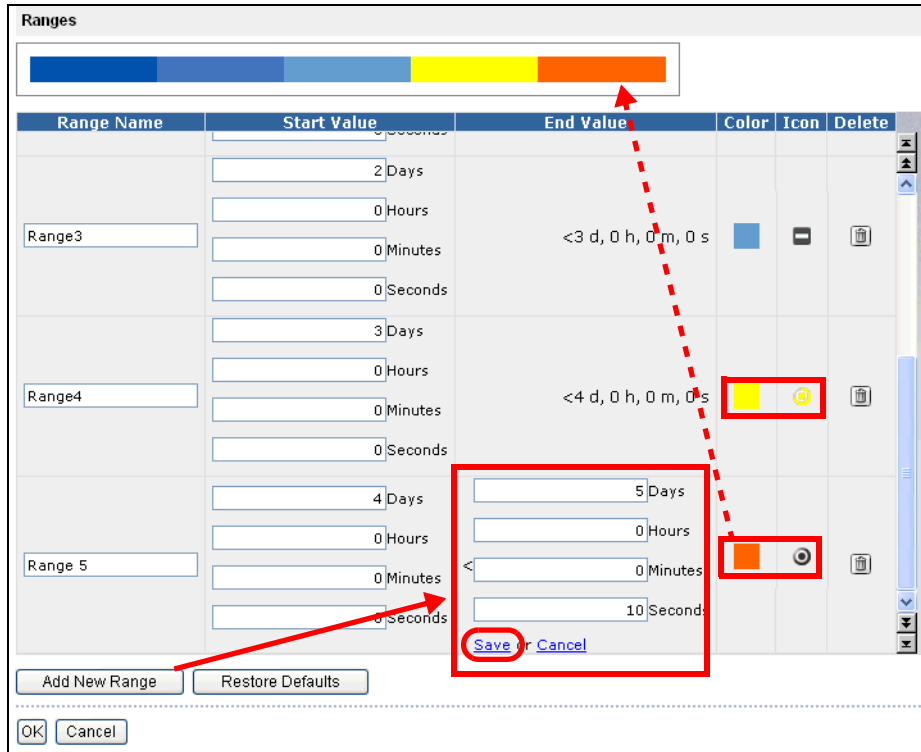


Figure 14-76 KPI range edit and color choice

- Click *Add New Range*. In the range field specify an upper limit of 5 days. Then click *Save*.
 - Select the color beside the range to edit it ■.
 - Select an icon to be displayed when that range is reached .
 - Click *OK* when finished.
- Configure the colors in the Shipped Order % view. The result is shown in Figure 14-77. The selected colors are orange (target not reached, yellow (close to target), green (target reached), and blue (even better).

KPI Name	Status	Value	Target	Value in Range
Average Shipped Order Fulfillment Time	-	-	3 d, 0 h, 0 m, 0 s	
Shipped Order %	-	-	90	

Figure 14-77 KPI view configured

Gauge view configuration

The Gauge view configuration is similar to the KPI view configuration. The only difference is the layout of the gauges, which can be horizontal, vertical, or in a grid.

To configure the layout of the Gauge view, follow these steps:

- ▶ In the Gauge view, click *Edit* (icon).
- ▶ The page displayed gives the ability to change the Gauge layout between horizontal, vertical, and grid (Figure 14-78).
- ▶ Select all the gauges to be displayed and click *OK*.

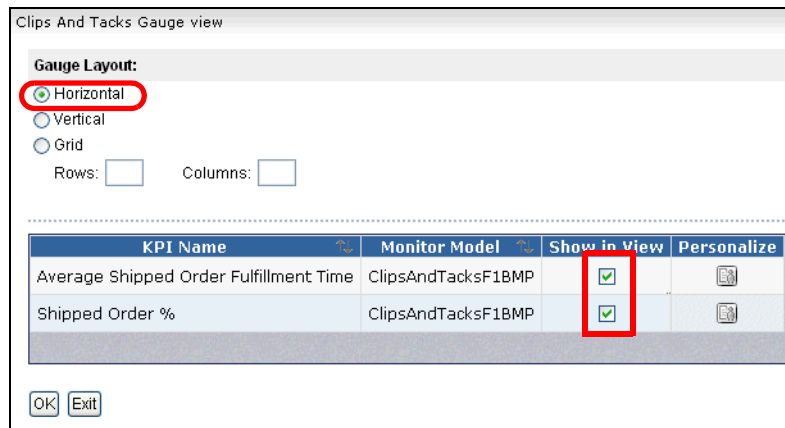


Figure 14-78 Configuration of gauges view

- ▶ Click *Personalize* for each view and define ranges and colors in the same way as for the KPI view. The finished view is shown in Figure 14-79.

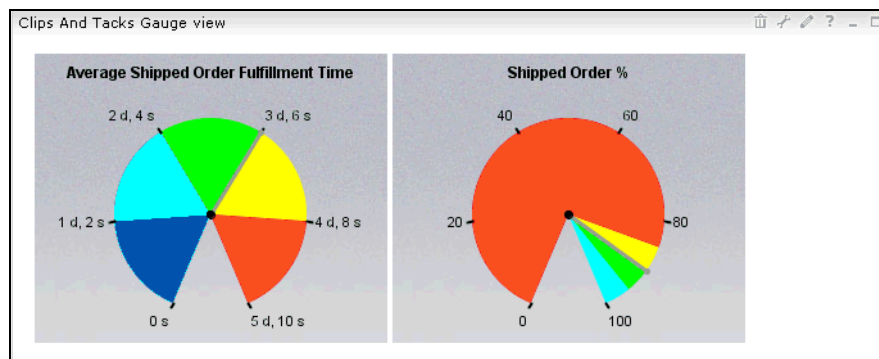


Figure 14-79 Gauge view configured

Dimensional view configuration

In the dimensional view, we configure which measures are displayed against the location dimension.

- ▶ Click *Configuration* (icon).
- ▶ Select the monitor model (ClipsAndTacksF1BMP) and the cube (ORDERHANDLINGFUTURE1 CUBE) for dimensional analysis.
- ▶ A list of the available dimensions is displayed (Figure 14-80).
- ▶ Select *Measures* and click *Add>* to the row dimension.
- ▶ Select *Location* and click *Add>* to the column dimension.

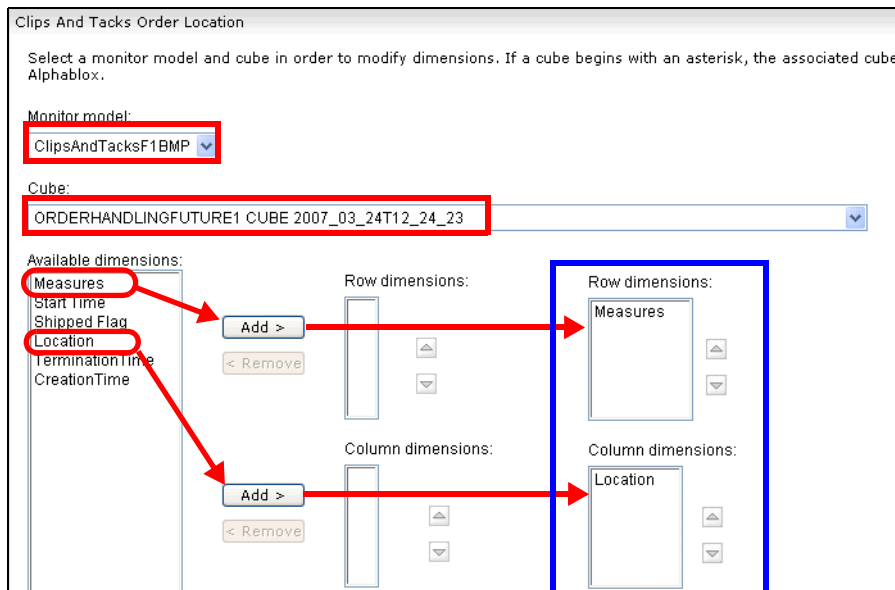


Figure 14-80 Configuring the dimensions view

- ▶ Click *Next* to tailor the view (Figure 14-81). If you already have instances (processed orders), then the instances count would show the value. If you configure before having orders, then the count is 0.

Note: Further configuration of the dimensions view may be better done after you have a number of order instances. We only show a few steps here.

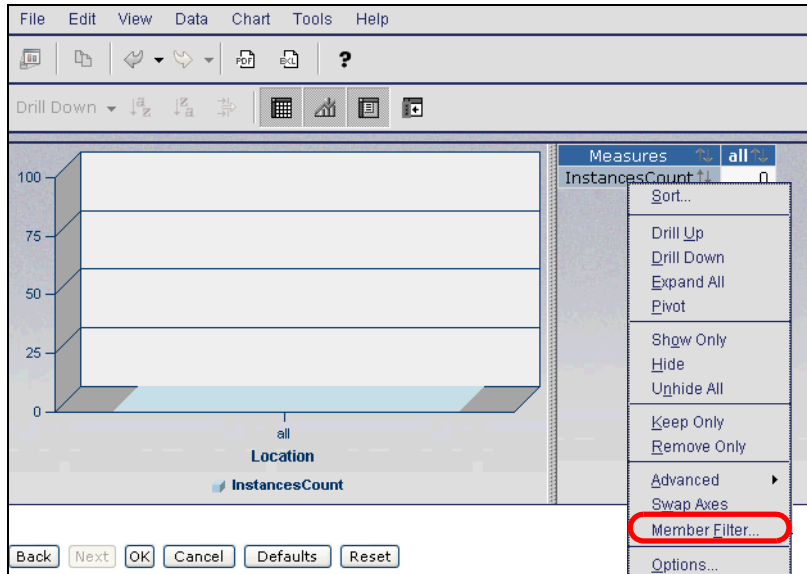


Figure 14-81 Dimensions view: Initial

- ▶ By default, the instances count is the only measure provided. To filter on the different measures that are modeled in the same cube, select *InstancesCount* and *Member Filter*.
- ▶ Select the measures to be displayed, and remove the instances count (Figure 14-82).

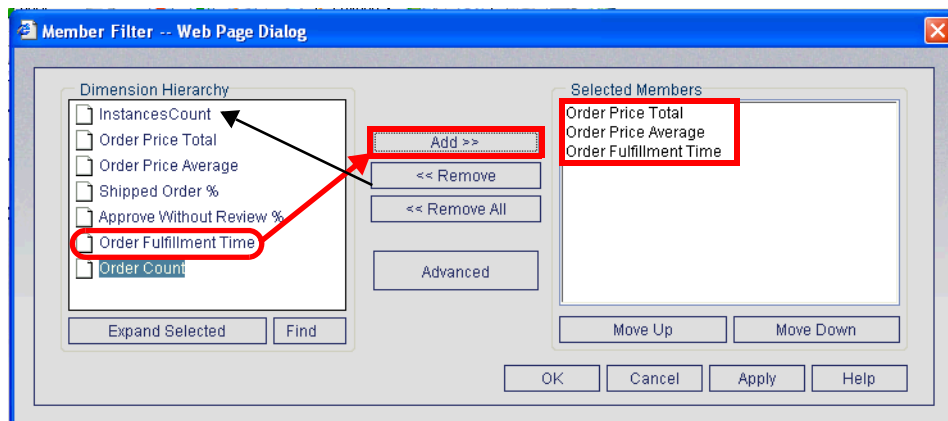


Figure 14-82 Selecting measures for dimensional analysis

You have to think about the value ranges that are displayed on one page. Order price and average may use the same scale, but the fulfillment time (milliseconds) is much higher. We can tailor the display to have two scales:

- ▶ Right-click the chart area and select *Chart Types*.
 - On the Chart Types tab, select *Vertical Bar, Side-by-Side, Dual Axis* (Figure 14-83).
 - On the Axes Placement tab, select *2nd y-axis* for the fulfillment time.

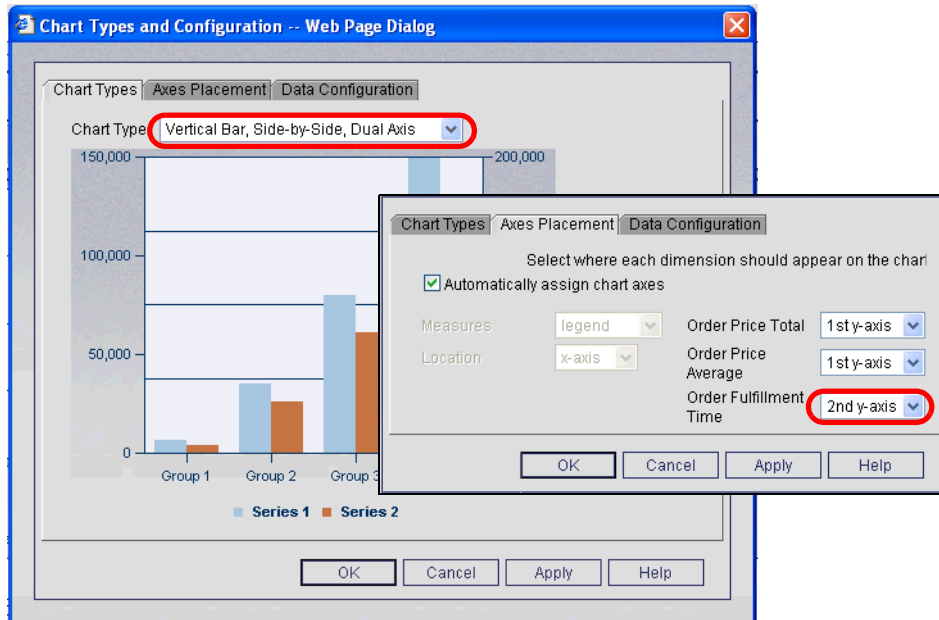


Figure 14-83 Selecting chart type and axis placement

- ▶ We make a correction to the way the duration is displayed in the grid. Instead of being displayed in milliseconds, we configure it to be displayed in hours.
 - Under Measures, select *Order Fulfilment Time* and *Advanced* → *Format Mask*.
 - To change the format from milliseconds to hours, select *Custom*, and for the mask type in the Mask field, type `0.00/3600000 'hours'` (Figure 14-84). For days, you could use `0.00/86400000 'days'`.

Format Mask Name (required):
Measures=Order Fulfillment Time

Format Type
Custom

Mask
0.00/3600000 'hours'

Example: 0.34 hours

Format mask will be applied to cells using the scope(s):
1: {Measures:Order Fulfillment Time}

OK Cancel Apply Help

Figure 14-84 Formatting the duration into hours

Alerts view configuration

The Alerts view only provides personalization facilities for editing the colors of the read and unread messages, the refresh rate, and the number of messages to be displayed per page (Figure 14-85).

Mark	Selected	Palette
Read	Black	Palette icon
Unread	Red	Palette icon

Auto refresh rate: 60 seconds

Number of rows to display: 10

Show situation event

OK Preview Exit

Figure 14-85 Editing the Alerts view

Export actual values view configuration

This view only has a configuration option. By clicking *Configuration* (icon), the list of models is displayed. Select the model you want to export data (Figure 14-86).

Clips And Tacks Export actual value


Select the monitor models a user will be able to export from.

Monitor models:
 ClipsAndTacksF1BMP

Figure 14-86 Actual values configuration

Authorization

Typically different user groups or individual users would be authorized to see different portal pages, and therefore different portlets.

For the ClipsAndTacks portal page, you set permissions by clicking *Set Page Permission*  (see Figure 14-63 on page 462).

- In the dialog, click *Edit Role* for User (Figure 14-87).

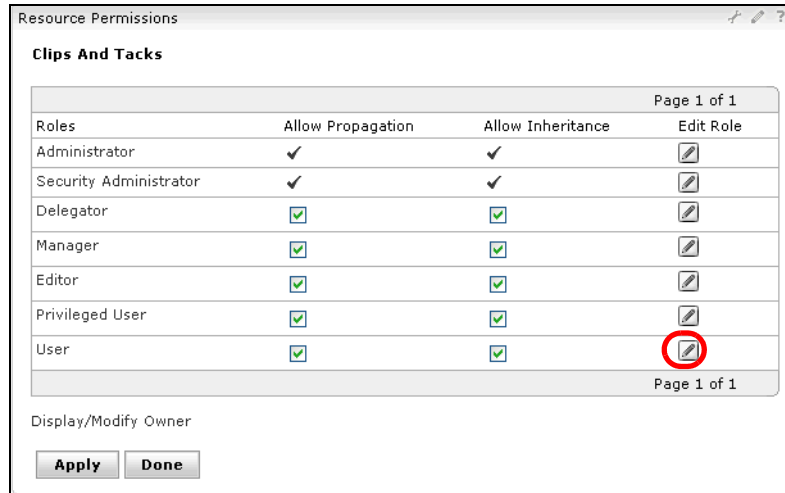


Figure 14-87 Defining portal page permissions

- Click *Add* to add groups or users to the role members (Figure 14-88).

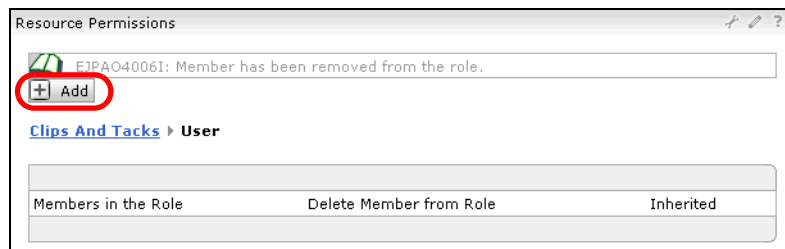


Figure 14-88 Adding groups and users to a portal page

- Search for all available user groups. Select the appropriate groups and click *OK* (Figure 14-89). You could also search for users by user ID, name, and so forth.

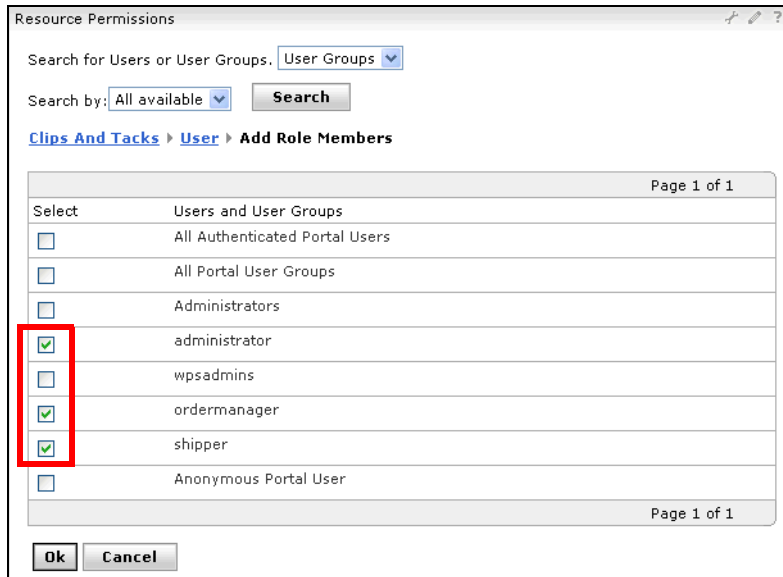


Figure 14-89 Selecting groups for a portal page

- ▶ Click **OK** and the selected groups are added to the role members (Figure 14-90).

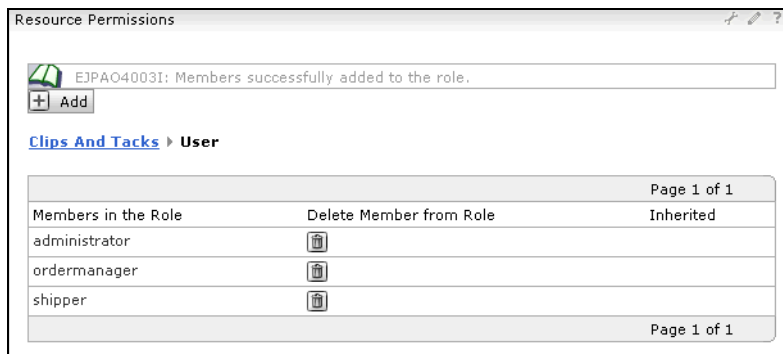


Figure 14-90 Portal page with permissions added

The ClipsAndTacks portal is now configured (Figure 14-91).

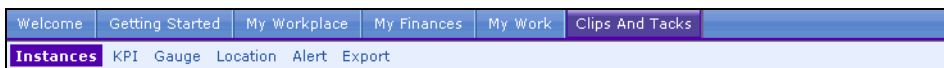


Figure 14-91 Portal pages for ClipsAndTacks

Measuring the Future 1 application

Our test environment is not a real ClipsAndTacks application where numerous customers submit their orders. To gather some meaningful results in the Monitor, we submit a set of orders and process them through the ClipsAndTacks system using the human tasks for processing of the orders.

Figure 14-92 shows an extract of the list of orders submitted:

- ▶ A small amount is under \$750 and is automatically approved by the business rule.
- ▶ A big amount is over \$750 and requires the order manager for approval.
- ▶ An amount between about \$500 and \$750 for customer 22222 is automatically approved, but does not pass the credit check and therefore also goes to the order manager.
- ▶ We submit 31 orders. 27 orders are shipped, and the other 4 orders are declined.

Order	Customer	Amount	AutoApprove	MgrApprove	Ship-When	Notif-When
1	12345	small	yes		2 days	
2	11111	big	no	yes	3	
3	22222	small	yes		2.5	
4	33333	small	yes		4	
5	44444	small	yes		2.5	
6	12345	big	no	NO		2.5
7	11111	small	yes		2.5	
8	22222	740	yes	yes	3	
9	33333	big	no	yes	4	
10	44444	small	yes		3	

Figure 14-92 Extract of list of orders submitted

Human tasks

We processed the human tasks by claiming the process instances, then waiting about 15 minutes before processing. This resulted in a measured processing time in the Monitor.

We waited before shipping the order between one and five days to get an average of just over three days.

The data was designed to give measurements that do not meet the expectations of ClipsAndTacks management, so that the process must be improved.

Tracking the orders in the Monitor databases

As described in “Monitor databases” on page 54, the Monitor uses two databases for reporting, the MONITOR database and the DATAMART database.

The most interesting data is in the DATAMART database in the table FCT_ORDERHANDLINGFUTURE1. In addition, there is a table for each monitoring context (one per activity). You can use the DB2 Control Center to view the instances in the two databases.

Verifying that DB2 replication is working

Replication between the MONITOR and DATAMART database can be verified by looking at the contents of the DATAMART fact tables in the DB2 Control Center:

- ▶ Open the DB2 Control Center.
- ▶ Navigate to *All Databases* → *DATAMART* → *Tables* and select the table associated with the parent monitoring context (Figure 14-93):
 - Name: FCT_ORDERHANDLINGFUTURE1
 - Schema: WBI

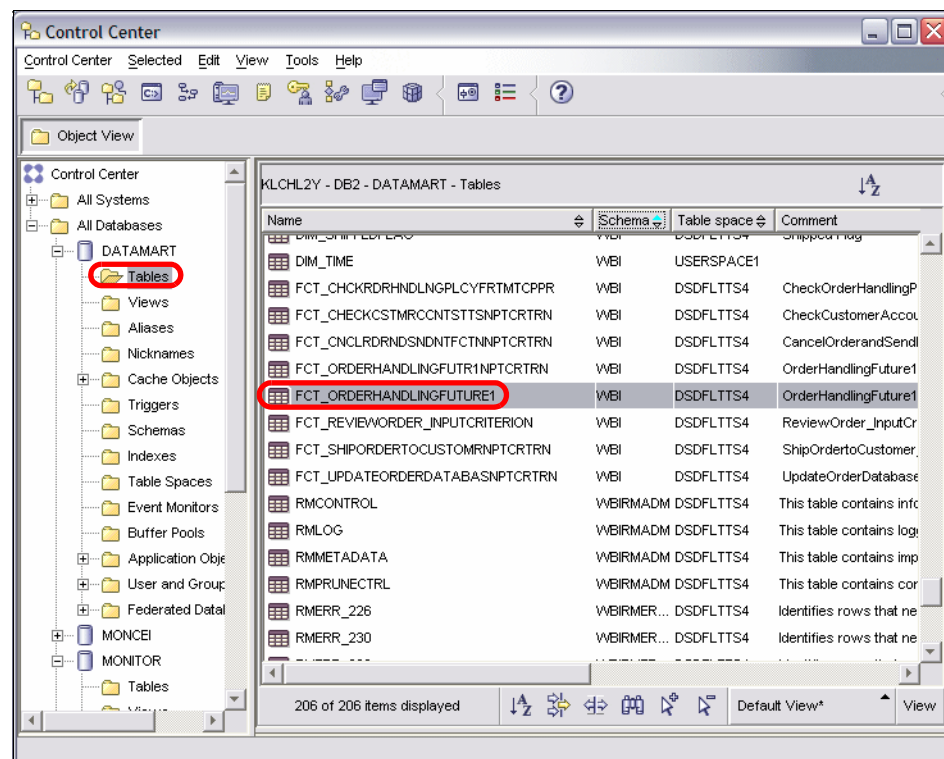


Figure 14-93 DB2 Control Center: Verify data replication

- Verify that the table has been populated (Figure 14-94).

MCI_MCIID	PARENT_MCIID	MCI_MCVID	M_PROCESSINSTANCEDID
2000		1172505420	_PI:90030111.194e78fd.eb...
2033		1172505420	_PI:90030111.194fe771.eb...
2038		1172505420	_PI:90030111.19503bac.eb...
2115		1172505420	_PI:90030111.1953bc18.eb...
2084		1172505420	_PI:90030111.19526070.eb...
2075		1172505420	_PI:90030111.1951ffa3.eb...
2056		1172505420	_PI:90030111.1951110d.eb...
2066		1172505420	_PI:90030111.19518c67.eb...
2093		1172505420	_PI:90030111.1952c7a6.eb...
2102		1172505420	_PI:90030111.19533b8e.eb...
2106		1172505420	_PI:90030111.19536cdf.eb...
2124		1172505420	_PI:90030111.19541bfb.eb...
2134		1172505420	_PI:90030111.19548592.eb...
2024		1172505420	_PI:90030111.194f6cf2.eb...
2138		1172505420	_PI:90030111.1954ba2f.eb...
2129		1172505420	_PI:90030111.1954501b.eb...
2119		1172505420	_PI:90030111.1953e4be.eb...
2110		1172505420	_PI:90030111.195395c4.eb...
2097		1172505420	_PI:90030111.1952ccdf.eb...
2088		1172505420	_PI:90030111.19529b37.eb...
2079		1172505420	_PI:90030111.19523142.eb...
2070		1172505420	_PI:90030111.1951bf7d.eb...
2061		1172505420	_PI:90030111.195157da.eb...
2052		1172505420	_PI:90030111.1950df7d.eb...
2047		1172505420	_PI:90030111.19509d06.eb...

Figure 14-94 DB2 Control Center: Verify data replication

If data is not yet present, close the window displaying the table, select *All databases* → *DATAMART* → *tables*, and click *Refresh* (context menu).

Understanding the tables in the Monitor databases

To understand the tables in the databases, documentation is generated when you generate the database scripts in “Step 1: Run Data Services Generation” on page 420. The generated folder contains two HTML files that describe the tables:

```
stateMapping.html
datamartMapping.html
```

Analyzing the Future 1 dashboard data

After running a number of orders through the business process, we can start analyzing the accumulated data and measure the Future 1 process. We go through the different views.

Instances view analysis

The Instances view provides the capability of displaying instances for your processes. For this model, a process instance represents an order, so each record in the instance view contains the metrics collected for an order.

The instances view is used to display the shipped orders and the orders that are still active (Figure 14-95).

Order City	Order Country	Shipped Flag	Order Price	Start Time	Order Process Time
Buffalo	USA	SHIPPED	79	Mar 3, 2007 11:40:59 AM	3 d, 1 h, 24 m, 55 s
Buffalo	USA	SHIPPED	150	Mar 3, 2007 11:39:45 AM	4 d, 1 h, 41 m, 33 s
Buffalo	USA	SHIPPED	2,495	Mar 3, 2007 11:44:14 AM	3 d, 1 h, 20 m, 30 s
Buffalo	USA	SHIPPED	150	Mar 3, 2007 11:37:20 AM	5 d, 31 m, 5 s
Buffalo	USA	SHIPPED	79	Mar 3, 2007 11:43:00 AM	3 d, 1 h, 22 m, 16 s
Buffalo	USA	SHIPPED	79	Mar 3, 2007 11:43:47 AM	3 d, 1 h, 21 m, 10 s
Buffalo	USA	SHIPPED	2,495	Mar 3, 2007 11:42:07 AM	2 d, 13 h, 5 m, 5 s
Chicago	USA	SHIPPED	2,495	Mar 3, 2007 11:39:20 AM	2 d, 50 m, 49 s
Chicago	USA	SHIPPED	2,495	Mar 3, 2007 11:42:49 AM	2 d, 13 h, 4 m, 42 s
Chicago	USA	SHIPPED	79	Mar 3, 2007 11:40:33 AM	3 d, 1 h, 25 m, 28 s

Figure 14-95 Instances view, shipped orders


In the Instances view the data of the metrics captured for each order is displayed:

- ▶ Information about the location of the shipment (city and country).
- ▶ The shipped flag, indicating whether the order is shipped.
- ▶ The order price represents the total price of all the items inside the order.
- ▶ The start time represents the time when the order was issued.
- ▶ The order process time is the total elapsed time until the order is shipped.

You can reconfigure the view to show active instances only (Figure 14-96).

Order City	Order Country	Order Price	Start Time	Order Process Time	Order
Buffalo	USA	2,495	Mar 5, 2007 9:41:06 AM	4 d, 1 m, 7 s	
Buffalo	USA	2,495	Mar 5, 2007 9:40:14 AM	4 d, 1 m, 59 s	
Markham	Canada	2,495	Mar 5, 2007 9:41:21 AM	4 d, 1 m, 45 s	

Figure 14-96 Active instances

At the end of each row there is a column for each activity of this process and there is an arrow icon . When this icon is clicked, it will drill down to display the metrics for that activity.

Key performance indicator view analysis

Key performance indicators represent an aggregated type of data across the entire set of instances, whether average, sum, count, maximum, or minimum.

The ClipsAndTacks KPI view is shown in Figure 14-97.

KPI Name	Status	Value	Target	Value in Range
Average Shipped Order Fulfillment Time	🟡	3 d, 2 h, 34 m, 18 s	3 d, 0 h, 0 m, 0 s	
Shipped Order %	🟡	87.10	90	

Figure 14-97 Key Performance indicator view

- ▶ The average order processing time for shipped orders is about 3 days and 2.5 hours, which is longer than the target of three days. The bar in the value range ends in yellow and the status icon is yellow as well.
- ▶ The percentage of shipped order for the previous 30 days shows about 87%, which is below the target of 90%.

Gauge view analysis

This view is another means of displaying the data that was seen in the KPI view (Figure 14-98). The arrows point to the current KPI value; both are not reaching the target (grey line at 3 days and 90%).

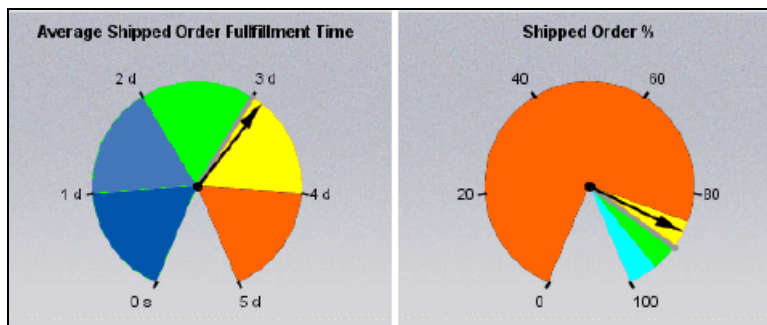


Figure 14-98 Gauge View

Alerts view analysis

Alerts are displayed based on the user logged onto the portal. If an alert is configured for a specific user or group, the alert is only displayed when this user (or a member of the group) is logged onto the portal.

<input type="checkbox"/>	Date and Time	Subject	Alert Source
<input type="checkbox"/>	Mar 7, 2007 3:16:45 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 7, 2007 3:47:34 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 7, 2007 4:00:59 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 7, 2007 4:07:07 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 8, 2007 12:17:37 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 8, 2007 12:17:37 PM	Order exceeded target shipping time	
<input type="checkbox"/>	Mar 8, 2007 12:17:37 PM	Order exceeded target shipping time	

Figure 14-99 Alerts view

The data displayed in the Alerts view is the date that the alert was created and the subject of the alert. As shown in Figure 14-99, there are orders that have exceeded the target shipping time.

Click the subject to display the details of the alert, as configured in “Configuring the Adaptive Action Manager” on page 441. The body of the alert contains data associated with the running process. When we configured the alert template configuration, we extracted the order number for the order that exceeded the target shipping time and added that to the body of the alert.

In the Alert Details view, you can also display the situation event and its details (Figure 14-100).

Clips And Tacks Alerts

Details:

Date and time:

Subject:

Business situation name:

Body:

Situation Event:

CommonBaseEvents

Figure 14-100 Alert Details with message body

Dimensional view analysis

Data displayed in this view represents the measures, filtered, based on the dimensions that were created at modelling time.

For ClipsAndTacks, we defined a Location dimension and a set of measures. This enables us to identify the values of the measures, based on the location (Figure 14-101).

The data in the Dimensions view is divided into:

- ▶ Grid (right side):
 - In the grid view, we have the data displayed based on the different location, for example, Canada and USA.
 - Data could be drilled down to be displayed by the cities within each of these countries.
 - For each country we can analyze the total and average order price, and the order fulfillment time
- ▶ Graph (left side):
 - We defined the order fulfillment time to be on a different scale. This second scale is shown on the right-hand y-axis.

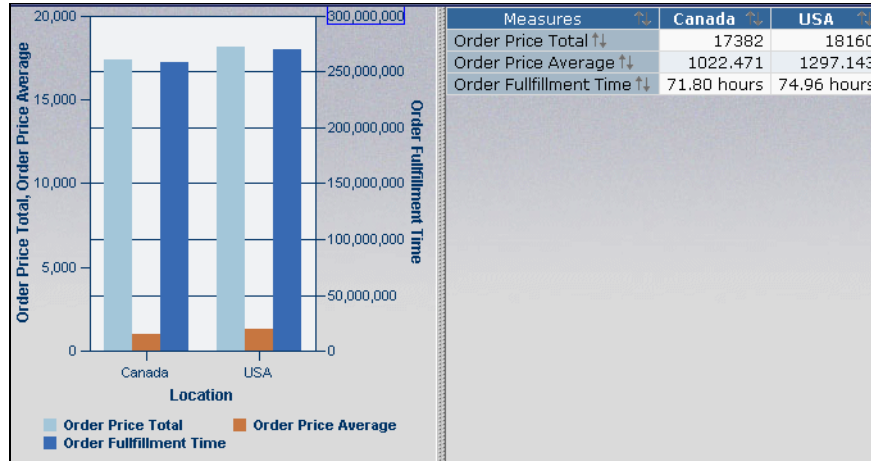


Figure 14-101 Dimension view: Order data by location

How to work with the Dimensions view is described in more detail when we measure the Future 2 process. Refer to “Dimensions view” on page 556 for examples of drill-down and selection of measures to be displayed.

Other views, and wiring views together

For the Future 2 process, we configured additional views and also show how to wire two portlets together in one view. Refer to “Monitor Dashboard for the Future 2 application” on page 555.

Database replication problem

By default, database replication only goes back for seven days. If you do not have the replicators running for seven days, then you get an error at the start of the capture replicator:

```
ASN0121E CAPTURE "CAPTURE_141" : "WorkerThread". The Capture program warm start failed because existing data is too old. The Capture program will terminate.
```

You can overwrite the default time limit in the capture command files. See “DB2 replication scripts” on page 449 for the location of the command files. You have to edit the data capture command file (StartCapture_x.bat) and change the command:

```
db2cmd asncap CAPTURE_SERVER=STATE CAPTURE_SCHEMA=CAPTURE_xxx  
CAPTURE_PATH="C:\IBM\WebSphere\Monitor\rm\logs\s2r\capture" LAG_LIMIT=xxxx
```

The LAG_LIMIT is the time in minutes since the last capture run. If you know when you last ran the command, calculate the time in minutes as days*1440 (for example 20160 for 14 days), then change the command with LAG_LIMIT=20160.

The actual value when the command ran the last time can be found in the table CAPTURE_xx.IBMSNAP_CAPMON, column SYNCHTIME, in the MONITOR databases. To calculate the minimum number of minutes for the LAG_LIMIT parameter, you can issue this SQL statement against the table:

```
SELECT TIMESTAMPDIFF(4, CHAR(CURRENT_TIMESTAMP - MAX(SYNCHTIME))) as  
Minimum_Minutes FROM CAPTURE_xx.IBMSNAP_CAPMON;
```

Alternatively, you can issue a cold start of the capture commands, however, this will replicate all the data in the tables:

```
db2cmd asncap CAPTURE_SERVER=STATE CAPTURE_SCHEMA=CAPTURE_xxx  
CAPTURE_PATH="C:\IBM\WebSphere\Monitor\rm\logs\s2r\capture" STARTMODE=cold
```

You can find information about the database replication configuration in the Monitor Information Center (*Troubleshooting* → *Database services troubleshooting* → *Runtime issues*) at:

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp>

Summary

In this chapter we deployed the monitor model on the Monitor Server. We configured the Dashboard, and the Adaptive Action Manager such that we could use the Monitor to display data regarding our process.

We also described the measurement of the Order Handling (Future 1) process. Because our application is not a real application with many customers, we used a script to submit orders to get some meaningful results.

We used the Monitor Dashboard to view and analyze the results. These results will enable ClipsAndTacks management to take action to improve the key performance indicators.

Continuous process improvement

In Part 5 we describe how to improve the business process based on the measurements performed using WebSphere Business Monitor.

We compare the results of the Monitor with our simulation results in the Modeler. Then we change the process to improve the order turn-around time. We export the model once more and implement the new model in WebSphere Integration Developer. Finally, we deploy the model to WebSphere Process Server and monitor the new application.



Comparing the measurements and modeling the Future 2 process

This chapter describes how we use the measurements from the WebSphere Business Monitor to improve the business process.

We export measurement results from the Monitor and import them into the Modeler. After comparing the Monitor results with the Modeler simulation, we decide what changes should be made to the business process.

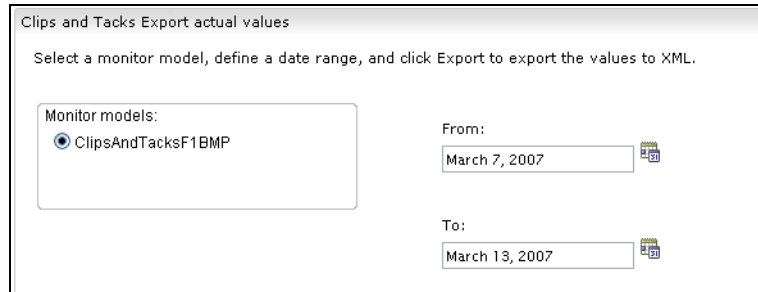
We implement the proposed changes and create the *Order Handling (Future 2)* process. We simulate and analyze the new process to predict the improvements.

One of the major changes is to add a customer classification of regular, silver, or gold, and use this classification to approve more orders automatically.

Exporting data from the Monitor

After completing the measurements of the Future 1 process, we export the data from the Monitor Dashboard into an XML file.

The activity duration and decision percentage that we measured are part of the ClipsAndTacksF1BMP monitoring model. In the Dashboard, select the export page and export the values for a certain time period (Figure 15-1).



Clips and Tacks Export actual values

Select a monitor model, define a date range, and click Export to export the values to XML.

Monitor models:

ClipsAndTacksF1BMP

From: March 7, 2007

To: March 13, 2007

Figure 15-1 Exporting monitor measurement data

When exporting the actual values of each monitor model, you are prompted for a location and name of the output file, for example:

```
SG247148\sampcode\monitor\export\ClipsAndTacks.xml
```

Let us study the file exported of the ClipsAndTacksF1BMP model. The output file is an XML file where the tags contain a tracking key that relates the data collected from the Monitor at runtime to the components modeled in the Modeler.

An excerpt of the XML file produced by the Monitor is shown in Figure 15-2. The values are not easily matched to the tasks in the Modeler process diagram. The internal tracking key is not visible in the Modeler diagram or Attributes view, they are exported to the Monitor in the Monitor model (.mm) file.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed Material - Property of IBM
5724-M24
(c) Copyright IBM Corp. 2005,2006. All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.-->
<runtimeData
  xmlns="http://www.ibm.com/xmlns/prod/websphere/monitoring/6.0.2/mr">
  <measure>
    <trackingKey>BLM-6647c2d7b35fab7b5f82e2272db98ea0
      /BLM-25b3c150ee68a024f8bc004192a2f3d3/outputSetProbability
    </trackingKey>
    <valueType>double</valueType>
    <value>55.17</value>
  </measure>
  ...
</runtimeData>

```

Figure 15-2 Excerpt from Monitor export XML file: ClipsAndTacks.xml

Importing data into the Modeler

We can import the Monitor output into a Modeler project and update the model with the measured values:

- ▶ In the Modeler, select the Order Handling (Future1) process and *Import*.
- ▶ Select *Monitoring result (.xml)*, then click *Next* (Figure 15-3).

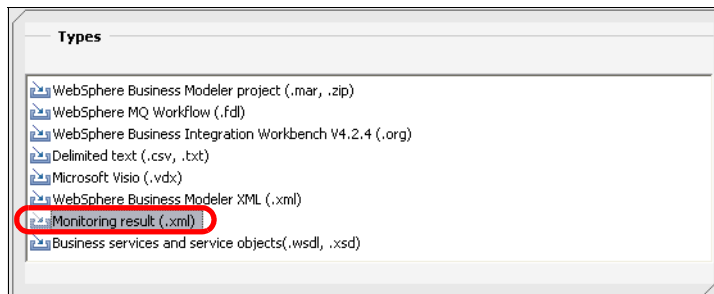


Figure 15-3 Modeler: Importing Monitor results (1)

- ▶ Select the `ClipsAndTacks.xml` file and click *Next* (Figure 15-4).

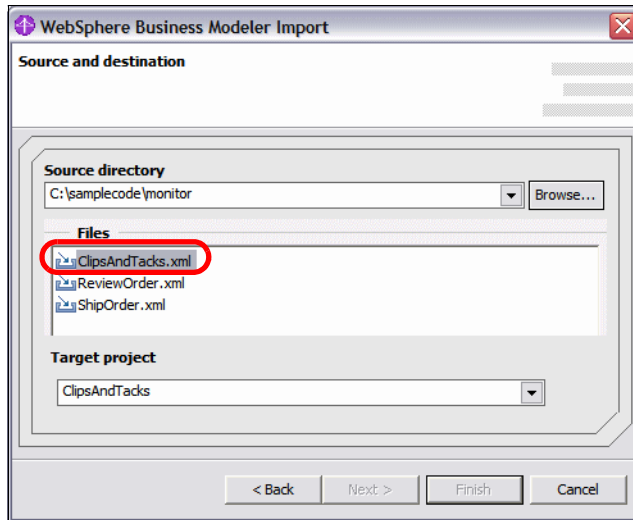


Figure 15-4 Modeler: Importing Monitor results (2)

- ▶ Click *Display Possible Updates* and the list at the bottom is filled. Click *Finish* (Figure 15-5).

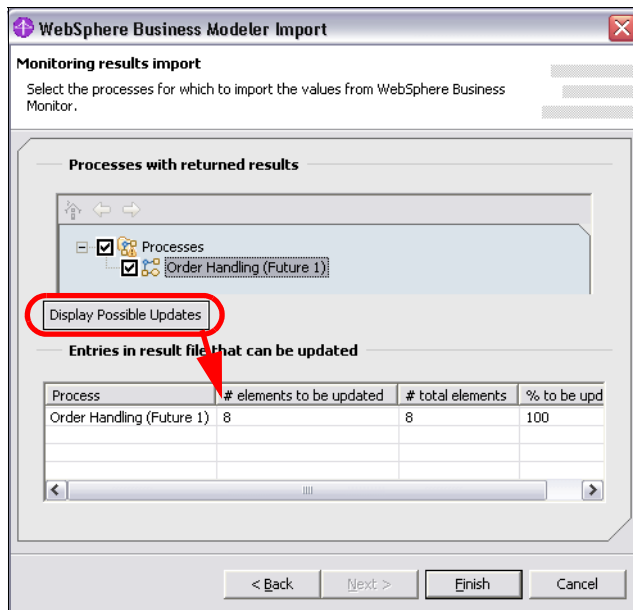


Figure 15-5 Modeler: Importing Monitor results (3)

- Processing starts and displays a dialog indicating that the import was successful (Figure 15-6). If there are problems, details can be inspected.

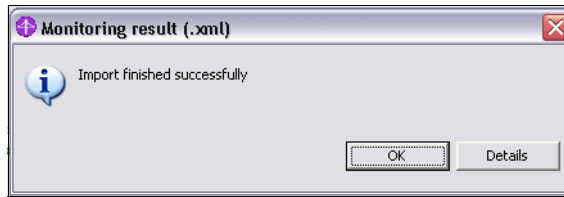


Figure 15-6 Modeler: Importing Monitor results (4)

For the decision percentages, we only created a monitor model measure for the Approve Without Review? Yes branch. Therefore, an error shows up after importing the actual values into Modeler (Figure 15-7).

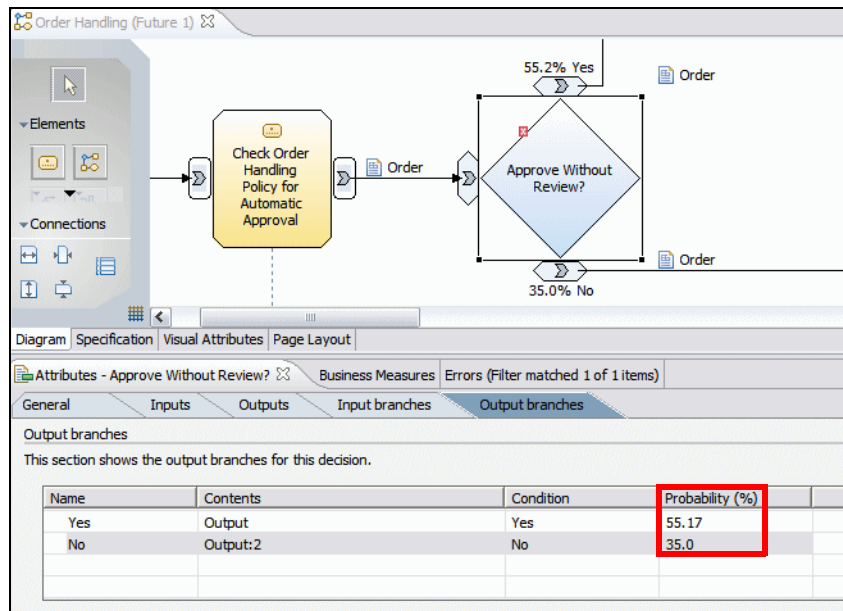


Figure 15-7 Decision error after import

To resolve the error:

- Select the decision in the diagram.
- In the Attributes view, Output branches tab, the values should add up to 100.
- Change the NO probability value to (100 - Yes probability value), in our case the YES probability is 55.17, so the NO probability value should be 44.83.
- Save the model after making the changes.

Analyzing the Monitor measurements

To analyze the results, we open the Order Handling (Future 1) process:

- ▶ We can see the new percentages for the decision blocks (Figure 15-8).

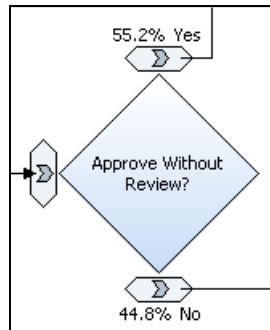


Figure 15-8 Decision block with new percentage values

- ▶ We can see the task duration by selecting the Duration tab in the Attributes view after selecting a task (Figure 15-9); for example, Check Order Handling Policy for Automatic Approval (1.509s) or Review Order (15m 1.481s).

The image shows two screenshots of the "Attributes" view in a software application, specifically the "Duration" tab. The top screenshot is for the task "Check Order Handling Policy for Automatic Approval". The "Processing time" section shows a "Specific value" dropdown and a table of time units: Days (0), Hours (0), Minutes (0), Seconds (1), and Milliseconds (589). The "Seconds" and "Milliseconds" fields are highlighted with a red box. The bottom screenshot is for the task "Review Order". It shows the same "Processing time" section with Days (0), Hours (0), Minutes (15), Seconds (1), and Milliseconds (481). The "Minutes", "Seconds", and "Milliseconds" fields are highlighted with a red box.

Figure 15-9 Activity duration measured by Monitor

Comparing the measurements with the simulation

First we analyze the decision percentages:

- ▶ We did not reach the 65% target automatic approval rate (55.2%).

Next we analyze the durations:

- ▶ We got good results for the non-human tasks and reached about 1 second for each task.
- ▶ We came close to the 15 minutes for the order manager (15m 1s).
- ▶ We exceeded the shipping time of 16 minutes (17m).

The measurements from the Monitor show that we still have too many orders being reviewed by the order manager, which slows down the process and leads to a longer total time that an order spends in the system.

KPI analysis

From the KPI measurements in the Monitor, we know:

- ▶ The percentage of shipped orders is only about 87% instead of 90%.
- ▶ The average duration exceeds the target of 3 days by a few hours.

Developing a strategy for improvements

Based on the Monitor measurements, the management of ClipsAndTacks decides to stream-line the business process in two ways.

Automatically approving more orders

To get more orders through the system without involvement of the order manager, the limit for automatic approval must be raised. However, it could be dangerous to raise the limit for all customers, especially new ones.

The solution is to introduce a customer classification system:

- ▶ Regular customers
- ▶ Silver customers
- ▶ Gold customers

Gold and silver customers have a higher limit for automatic approval of their orders. The business rule for automatic approval is changed as follows:

```
Order Handling Policy
=====
Default: Orders are reviewed by the system for automatic approval
=====
The order can be automatically approved without review if:
- the total order price is less than $750
- the total order price is less than $1250
  and the customer is classified as "Silver"
- the total order price is less than $1750
  and the customer is classified as "Gold"
=====
```

This new business rule does not change the flow of the order handling process. The new rule will be implemented in Integration Developer.

Automatically declining orders for customers with low credit rating

The checking of the customer account status sends orders back to the order manager for review. In most cases the order manager has declined the order based on the account status.

To get orders through the system faster, management decides that orders that do not pass the customer account status check are declined immediately. This will further reduce the involvement of the order manager.

However, this is a change of the flow in the order handling process.

Creating the Future 2 process in the Modeler

The best way to change the model and keep the old model as well is to work with a new project:

- ▶ Export the project as a WebSphere Business Modeler project (.mar) file.
- ▶ Import the exported file into a new project name Clips And Tacks Future 2.
- ▶ Rename the process from Order Handling (Future 1) to Order Handling (Future 2).

Note: You can import the new model from:

SG247148\samrcode\model\Clips And Tacks Future 2.mar

See "Importing the current process model using the Modeler" on page 83 for instructions on how to import a model.

Add classification to the customer record

The customer record is changed by adding the Classification (Figure 15-10):

- ▶ Open the Customer Record business item and click *Add*.
- ▶ Overtyping the name with Classification. The rest is fine.
- ▶ Click *Move Up* to move the item up two places.

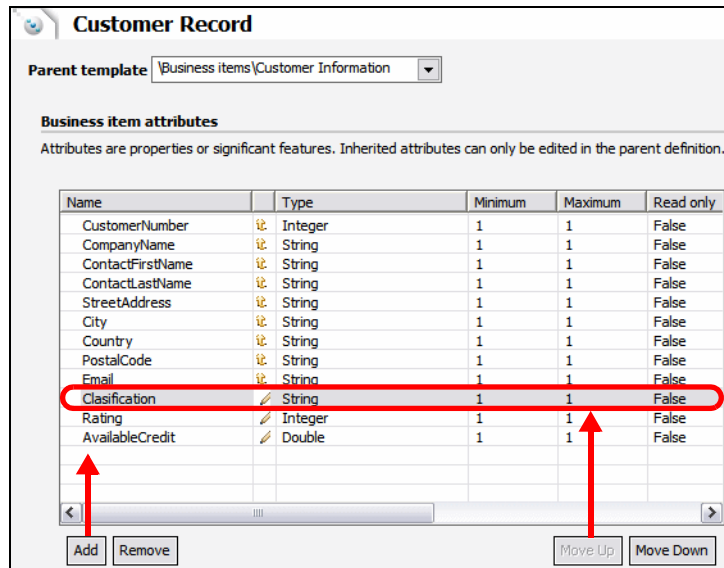


Figure 15-10 Adding customer classification

Change the business rule

The business rule is defined as an annotation (comment) in the Modeler. Therefore, change the annotation and you are done. The work will be performed in Integration Developer.

Change the percentage of automatically approved orders

Because of the customer classification (silver, gold), more orders will be automatically approved. For the simulation we have to change the percentages accordingly (Figure 15-11). We also change the percentages of the other decisions.

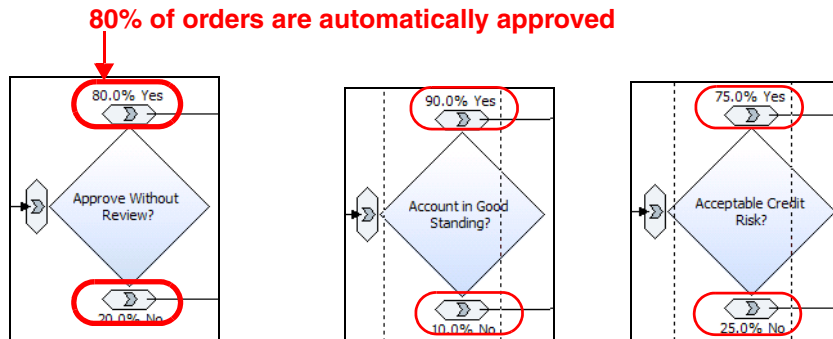


Figure 15-11 Decision percentages

Change the process flow

The Check Customer Account Status activity verifies the credit rating of a customer. Based on that a decision is made to approve the order. ClipsAndTacks management has decided to immediately decline an order if the credit check is negative.

Therefore we have to change the flow and route an order from Account in Good Standing to Cancel Order and Send Notification, instead of routing the order to the order manager (Review Order).

The changes to the model are shown in Figure 15-12:

- ▶ Remove the merge in front of Review Order.
- ▶ Connect Approve Without Review (No) directly to Review Order.
- ▶ Add a merge in front of Cancel Order and Send Notification.
- ▶ Connect both Acceptable Credit Risk (No) and Account in Good Standing (No) to the new merge. Note that you can grab the endpoint of a connection and move it to the merge input.
- ▶ Connect the output of the new merge to Cancel Order and Send Notification.
- ▶ Rearrange the diagram.

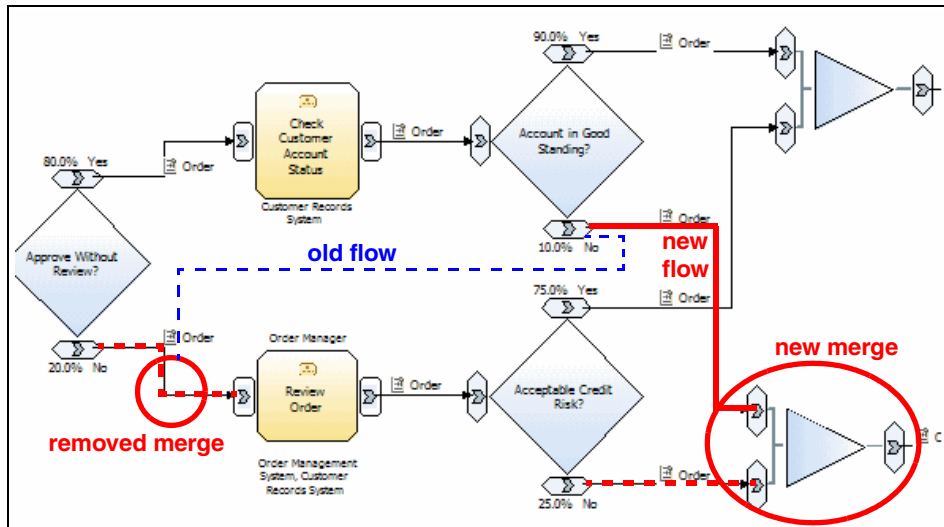


Figure 15-12 Changing the process flow

Changing the account status check to a global process

The Check Customer Account Status task is a local task. After discussions with the systems architect, who implemented the application in Integration Developer, the decision is made to change this task to a global process.

The reason for this change is to make the implementation more flexible, that is, easier to replace the implementation of this activity when required. We know that this activity uses an external Web service for the credit check. We do have the Web service interface available and can import it into Modeler.

To change the task to a global process, follow these steps:

- ▶ Select the Check Customer Account Status task and *Convert to* → *Global Process*.
- ▶ A warning is displayed that the conversions may result in errors or loss of information (Figure 15-13).
- ▶ Click *Preview* and read that the technical attributes will be lost.
- ▶ Click *OK*.

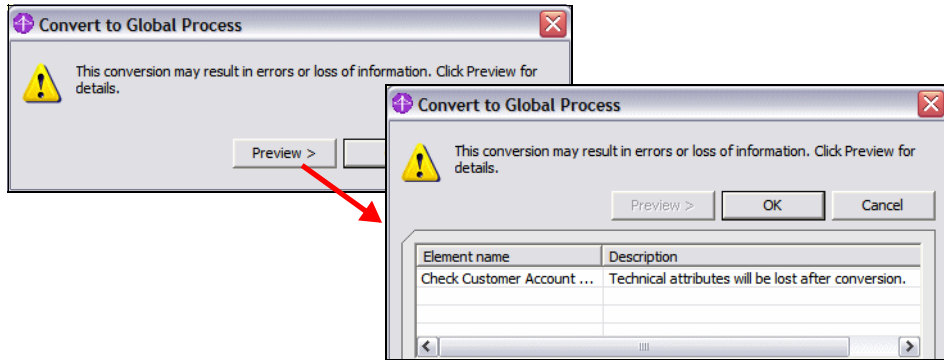


Figure 15-13 Changing a task to a process

- You can change the name if desired. Type Check Customer Credit as name and click *Finish*.

The task is now displayed with double lines in the diagram and also appears as a separate entry in the Project Tree view (Figure 15-14).

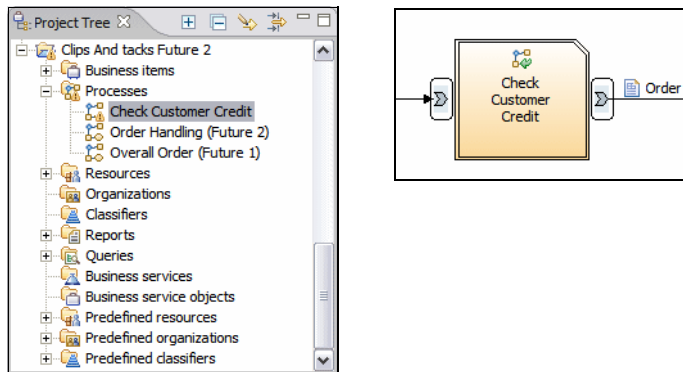


Figure 15-14 Project Tree view with global process

Order Handling (Future 2)

The updated model is shown in Figure 15-15.

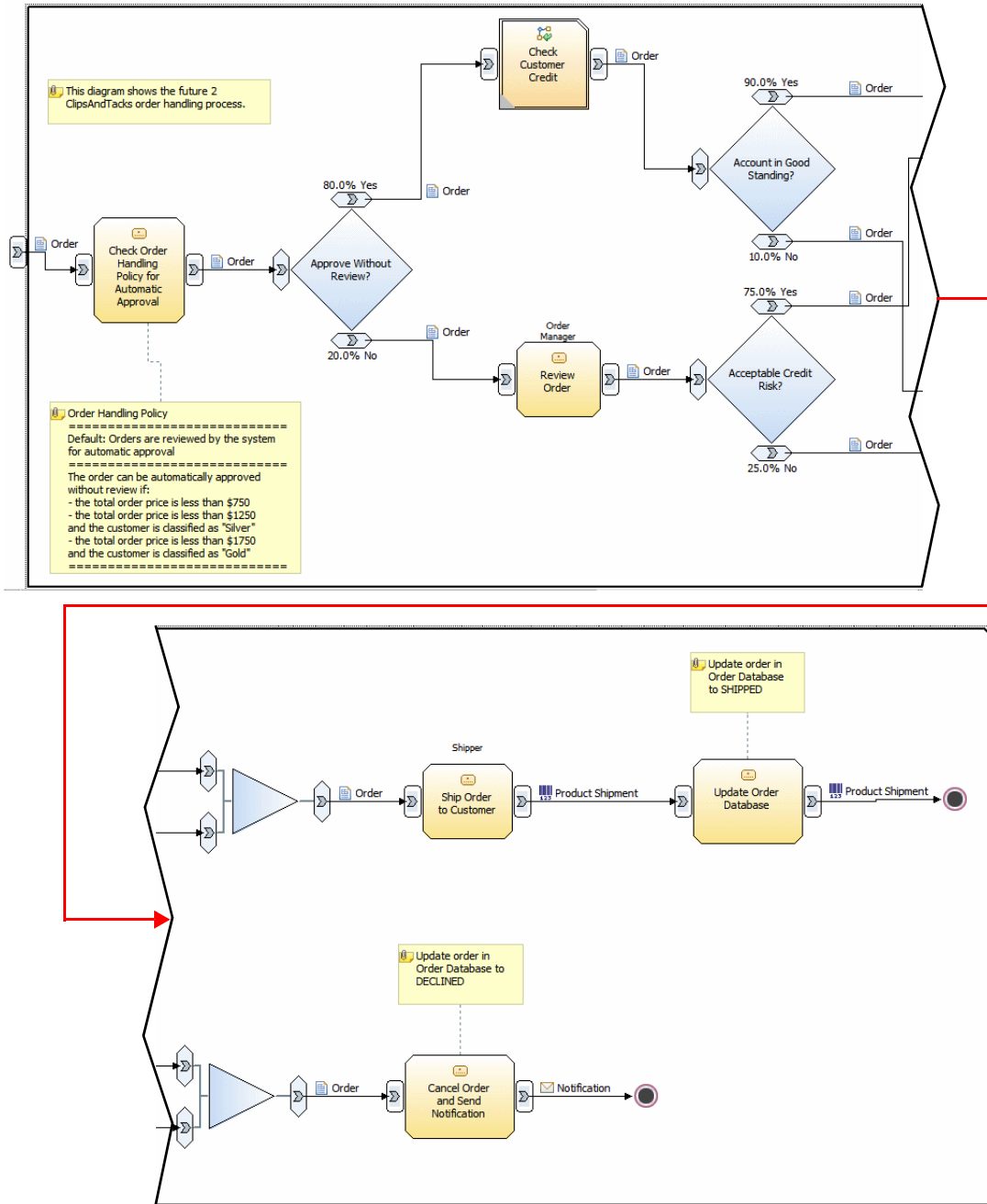




Figure 15-15 Order Handling (Future 2) process diagram

Modeling the global subprocess

The Check Customer Credit process takes an order as input, calls the external Web service, merges the result of the Web service into the order (actually into the customer record inside the order) and returns the order.

Preparation

Open the Check Customer Credit process. Delete the Start and Stop nodes, and reduce the size of the diagram using the *Decrease Horizontally*  and *Decrease Vertically*  icons.

Notice that the process has an input item and an output item associated (both are of type Order).

Importing the Web service

The application architect has provided the WSDL file of the external Web service:

```
SG247148\sampcode\model\webservice\CreditBean.wsdl
```

Import the Web service into the Clips And Tacks Future 2 project:

- ▶ Select the project and *Import*.
- ▶ Select *Business services and service objects (.wsdl, .xsd)* and click *Next*.
- ▶ Click *Browse* and navigate to the folder of the WSDL file. Select the folder, verify the target project, and click *Finish* (Figure 15-16).

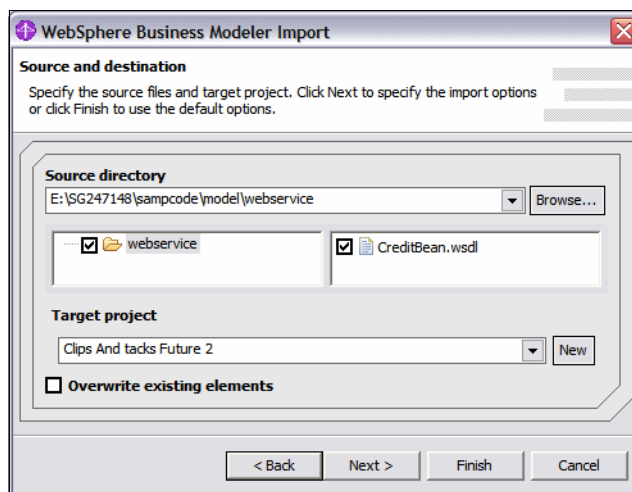


Figure 15-16 WSDL import

- ▶ The CreditBean appears in the Project Tree under Business services (Figure 15-17).

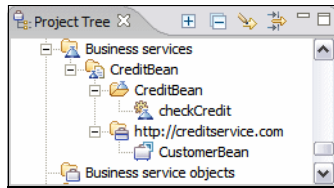


Figure 15-17 Project Tree with Business services

- ▶ Notice the operation name (`checkCredit`) and a data item (`CustomerBean`).
- ▶ Open the `checkCredit` operation. Select *Inputs* and *Outputs* to see that the input is the `CustomerBean` and that the output is an `Integer`. This output is the new customer credit rating (we are told by the architect).
- ▶ Open the `CustomerBean`. It contains two data items: `customerNumber` and `companyName`. These values are used by the Web service to access the customer's credit rating.

Creating the process flow

The input to the process is an order. To invoke the Web service, we have to extract the customer number and the company name. This is best done using a map.

We have to use the Web service output—the new customer credit rating—to calculate the available credit of the customer (inside the order). We can do this in a Java activity.

To create the process flow (Figure 15-18):

- ▶ Drop a Map from the Palette into the diagram. Accept the default name of Map.
- ▶ Drop the `checkCredit` Web service operation into the diagram.
- ▶ Drop a local task named `Update order with Customer Credit` into the diagram.
- ▶ Connect the process input to the map input. The `Order` is associated automatically.
- ▶ Connect the map output to the local task input. Associate the `Order` with this link.
- ▶ Connect the map output (another output) to the Web service input. The `CustomerBean` is automatically associated to the link.

- ▶ Connect the Web service output to the local task input (another input). An Integer is associated with this link.
- ▶ Connect the local task output to the process output. The Order is associated automatically.
- ▶ Add a Stop node and connect the local task to it (this is required for simulation to work).

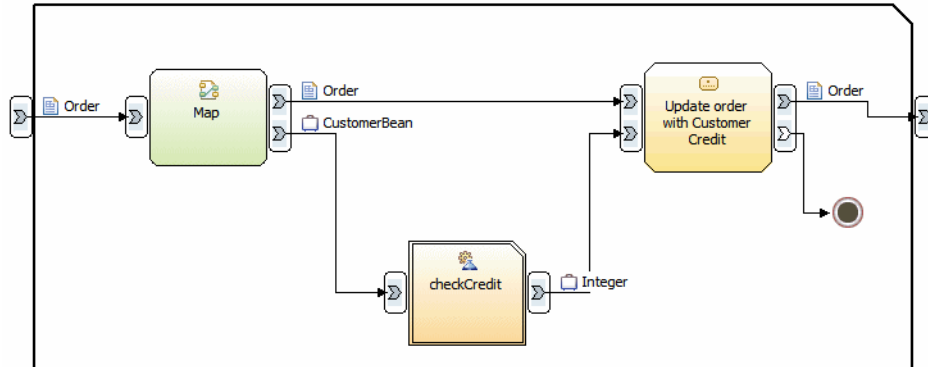


Figure 15-18 Check Customer Credit process diagram

Setting the attributes

For the simulation, we have to set durations:

- ▶ For the map, set the duration to 1 second.
- ▶ For the local task (Update Order with Customer Credit), set the duration to 1 second.
- ▶ To set the duration for the Web service, open the checkCredit operation from the Project Tree and set the duration to 5 seconds. Note that we could also assign cost and resources (for example, a Web service resource with availability).

Technical attributes: Implementation

Set the modeling mode to Process Server and set the implementation type of Update Order with Customer Credit to Java. The other activities do not have an implementation type.

Note: The process itself is of Request/Response type. It is invoked synchronously from the main process.

Deleting the overall process

We do not require the Overall Order (Future 1) process. You can either delete this process, or rename it to Overall Order (Future 2).

Simulating the Future 2 process

To simulate the process, we have to verify all the information of duration, cost, and availability. We will use the same attributes as in “Populating the simulation environment” on page 168.

Then we create a simulation snapshot and populate its attributes with the same values as in “Simulating the Future 1 process” on page 172:

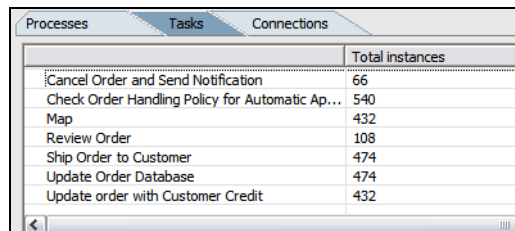
- ▶ Select the Order Handling (Future 2) process and *Simulate*.
- ▶ In the simulation profile set the random seed (1), total number of tokens (540), time table trigger (Online Request), frequency (5 minutes 20 seconds), maximum tokens per interval (270).

Be sure to limit the Resource Pool to one order manager and two shippers.

Finally, we can run the simulation and create results.

Analyzing the Future 2 simulation results

The Simulation Control Panel (Figure 15-19) shows that 474 orders out of 540 orders were shipped, which is approximately 88%. We see that 432 orders were automatically approved and went through the Web service. The Connections tab reveals that 41 of these 432 orders were declined because of the credit check.



Task	Total instances
Cancel Order and Send Notification	66
Check Order Handling Policy for Automatic Ap...	540
Map	432
Review Order	108
Ship Order to Customer	474
Update Order Database	474
Update order with Customer Credit	432

Figure 15-19 Simulation Control Panel Tasks

The process duration analysis (Figure 15-20) reveals that the average duration is just under 3 days, and that 87.78% of the orders are shipped.

Process Duration Simulation result Tuesday, March 13, 2007 8:03:38 AM PST Order Handling (Future 2) Tuesday, March 13, 20...				
Case Name	Distribution	Success Status	Average Elapsed Duration	Average Throughput
Case 1	72.41%	Succeeded	2 days 20 hours 7 minutes 1.592 seconds	0.01 work item / hour
Case 2	7.59%	Succeeded	10 seconds	360.00 work items / hour
Case 3	15.37%	Succeeded	5 days 9 hours 6 minutes 31.939 seconds	0.01 work item / hour
Case 4	4.63%	Succeeded	20 hours 45 minutes 19.599 seconds	0.05 work item / hour
All Cases			2 days 22 hours 7 minutes 38.84 seconds	0.01 work item / hour

Figure 15-20 Process duration results

- ▶ Case 1 orders are automatically approved and shipped.
- ▶ Case 2 orders are automatically approved but declined (credit check).
- ▶ Case 3 orders are approved (by order manager) and shipped.
- ▶ Case 4 orders are non-approved (by order manager) and declined.

These results show that it may be feasible to reach the KPIs.

Future 2 business measures

After the experience of specifying the business measures in Modeler and then having to redo most of the work using the Monitor Toolkit, we decided not to define any business measures in the Modeler for Future 2.

We tell the architect to implement the same KPIs and measures as for Future 1, with the addition of a new dimension:

- ▶ We want to use the customer classification as a dimension, so that we can analyze if our GOLD customer order more products—number of orders and average price of orders—than the SILVER and REGULAR customers.

Later we decided to export some simple business measures so that we had the SVG graphical model available for displaying KPIs and active instances in the graphical diagram (refer to “Using the visual model” on page 564).

Exporting for Integration Developer and Process Server

Now we are ready to export the revised process to Integration Developer.

Before exporting, verify that no errors are reported for the project when selecting *Modeling* → *Mode* → *WebSphere Process Server*.

To export the process for Integration Developer, select the Clips And Tacks Future 2 project and *Export* (context):

- ▶ Select the type of export you want. Select *WebSphere Process Server*. Click *Next*.
- ▶ Specify the target directory:
C:\SG247148\sampcode\model\export
- ▶ Select *Export entire project*. If you did not delete the Overall Order (Future 2) process, then select *Export specific elements* and select the two processes Check Customer Credit and Order Handling (Future 2).
- ▶ Specify the module name and interchange file name:
 - Select *Module project name* and enter: ClipsAndTacksF2
 - Select *Project Interchange name* and enter: ClipsAndTacksF2
- ▶ Click *Finish*.

Exported files

The result of the export is the ClipsAndTacksF2.zip file.

Summary

In this chapter we started with the continuous improvement cycle by taking the Monitor measurements back into the Modeler to improve the business process. We implemented the model changes, simulated and analyzed the model, and exported the model for deployment to Integration Developer and Process Server and for new measurements by the Monitor.



Implementing the Future 2 process using WebSphere Integration Developer

In this chapter we describe how to implement the different tasks of the Future 2 order handling business process. We follow a similar approach as in Chapter 10, “Developing the application using WebSphere Integration Developer” on page 211, but with a few modifications:

- ▶ We implement the business rule for automatic approval using a decision table.
- ▶ We use Information Services to access the database rather than Java.
- ▶ We use Visual Snippets to manipulate data in the business process.
- ▶ We implement the credit check as a Web service.
- ▶ We invoke the Order Handling process as a Web service and discuss the benefits.
- ▶ We investigate the impact of changing the Web service to an external Web service with a different interface by using maps.

Importing the Future 2 application

To import the revised application, follow the steps described in “Importing the model” on page 222. Select the **ClipsAndTacksF2.zip** interchange file exported from the Modeler. Select the ClipsAndTacksF2 project.

Note: The ClipsAndTacksF2.zip interchange file is available in the sample code for the book. It can be found in:

```
SG247148\sampcode\model\exportsolution
```

Wait until the workspace has been built and you should find these projects in the Web perspective:

- ▶ ClipsAndTacksF2App—Enterprise application
- ▶ ClipsAndTacksF2EJB—EJB project with session beans
- ▶ ClipsAndTacksF2Web—Web project (empty)
- ▶ ClipsAndTacksF2—Module with BPEL, tasks, rules, and data types

The Business Integration view after import is shown in Figure 16-1.

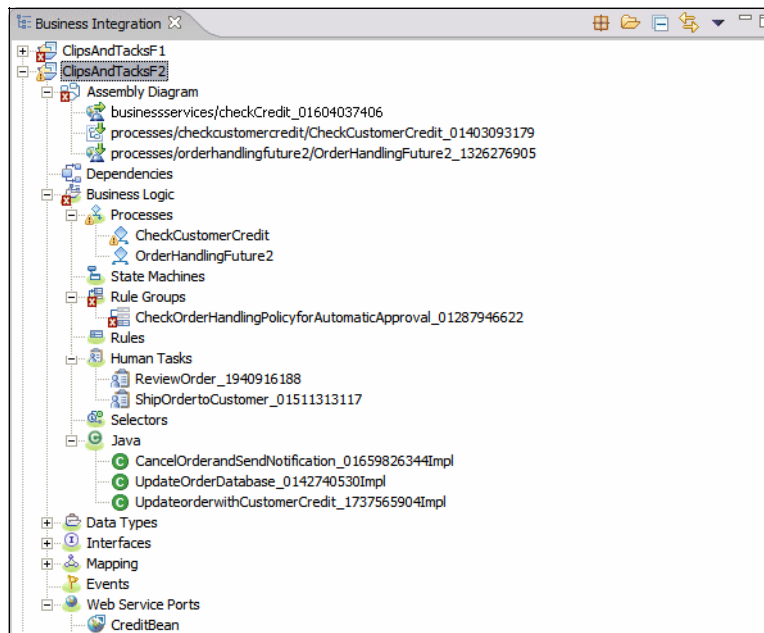


Figure 16-1 Business Integration view after import of Future 2

Note that we have two processes, each with an export. The Web service appears as a business service in the assembly diagram and as a port under Web Service Ports. We have two humans tasks and three Java activities. The rule group shows the typical error because we have no implementation. There is also an error in the assembly diagram that we have to investigate.

Assembly diagram

The assembly diagram is shown in Figure 16-2.

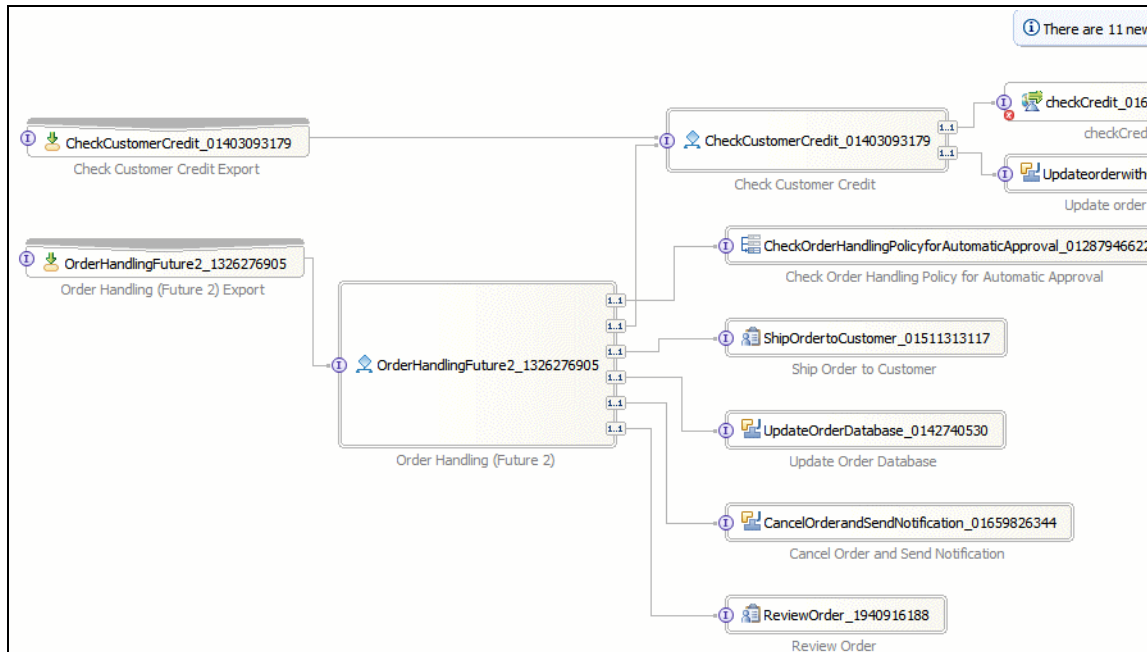


Figure 16-2 Order Handling (Future 2): Assembly diagram (compressed)

Notice the two exports: One export for the main process and one for the global process (Check Customer Credit Export).

The global process Check Customer Credit connects to the Web service and the Update order with Customer Credit activity.

Process diagram

The process diagram for order handling is shown in Figure 16-3.

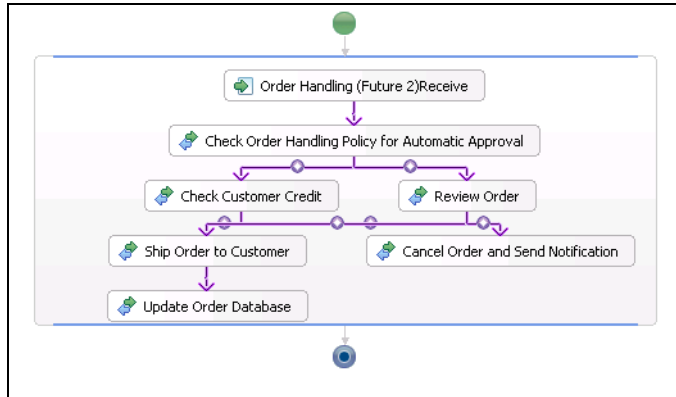


Figure 16-3 Order Handling (Future 2): Process diagram

The main process has no return value.

In the Properties view, leave *Automatically delete the process after completion* selected. We do not keep the finished processes.

The process diagram for the Check Customer Credit process is shown in Figure 16-4.

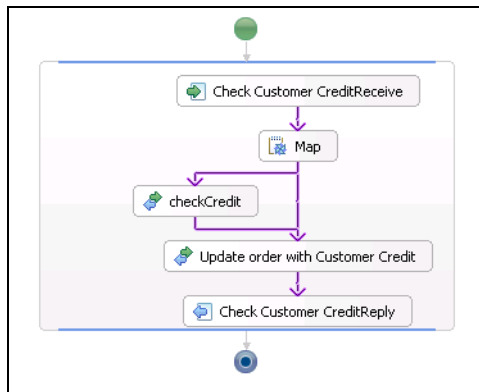


Figure 16-4 Check Customer Credit: Process diagram

The subprocess has a reply that goes back to the main process. Notice that the Map does not appear in the assembly diagram.

Completing the application in Integration Developer

To complete the Future 2 application, we follow steps similar to what we did in Chapter 10, “Developing the application using WebSphere Integration Developer” on page 211, with some changes to illustrate more functions of the process integration tools.

Updating assembly diagram

We have to change the namespace for the business items and remove the error. We also want to generate a Web service binding for the export for the Order Handling (Future 2) process, but we defer this to later.

Change namespace

We might still have active process instances for `ClipsAndTacksF1` in the Process Server in production. In this new version of the process we change the definitions of some of the business items. In order for both versions of the processes to be executing in the same environment at the same time, we have to change the namespace of the business items.

To change the namespace, follow these steps:

- ▶ In the Business Integration view, select the `ClipsAndTacksF2` module and expand the Data Types.
- ▶ Select one of the business items, `Order` for instance (not the `CustomerBean`), and select *Refactor* → *Change Namespace*.
- ▶ You will get a list of business items affected by the change. Click *OK*.
- ▶ Enter `http://BusinessItemsF2` as the new value (Figure 16-5).

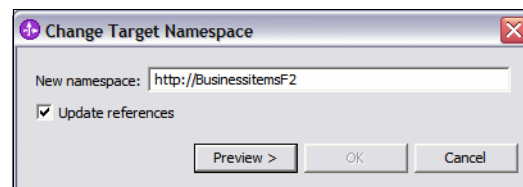


Figure 16-5 Change namespace

- ▶ Click *Preview* and then click *OK*.

The namespace has now been changed.

Specify bindings for the checkCredit import

Notice the red error mark in the checkCredit activity. We have to specify the binding information so that the Web service (the CreditBean) can be invoked.

- ▶ In the assembly diagram, select checkCredit_01604037406.
- ▶ In the Properties pane, select the Binding tab and click *Browse*.
- ▶ Select the CreditBean and click *OK* (Figure 16-6).

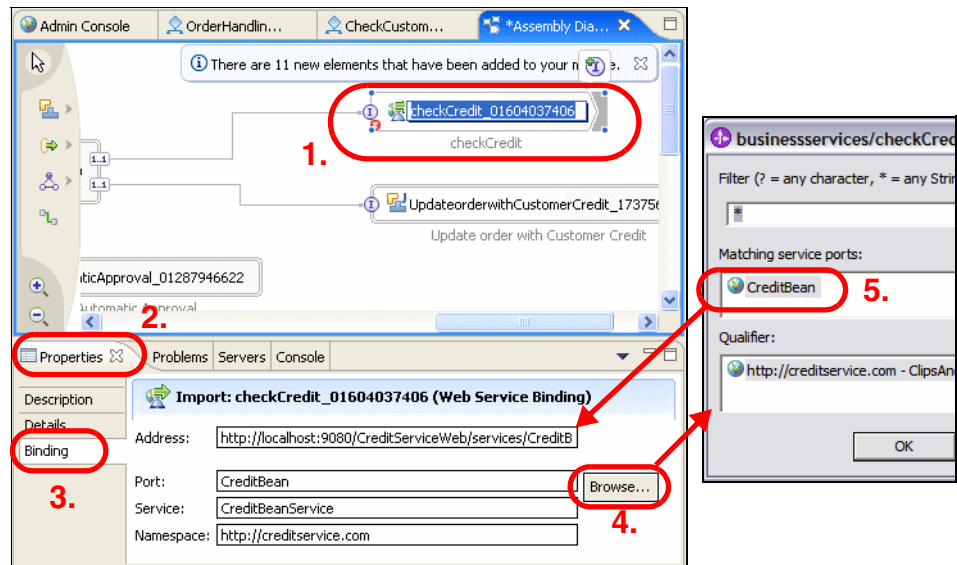


Figure 16-6 Specifying the Web service binding for checkCredit

- ▶ Save the assembly diagram and the error disappears. The import for checkCredit now knows the location of the Web service.

If the error does not disappear, you have to build the project by selecting the ClipsAndTacksF2 project and *Project* → *Clean*.

Fix invalid partner link

Open the Check Customer Credit process diagram. On the right-hand side, you may see that the CheckCustomerCreditPartner is marked with an error. This error does not always appear!

Select the invalid partner link and in the Properties view, Details tab, click *Browse*. Select the CheckCustomerCredit interface and click *OK*. Save and close the process diagram.

Implementing the business rule

The business rule for automatic approval was formulated in the Modeler, taking into account the classification of the customer. Now we have to implement the rule.

In the previous implementation we used a ruleset to implement the business rule. This time we will be using a decision table.

Tip: For a discussion on when to use a ruleset and when to use a decision table, read this article in the WebSphere Integration Developer information center:

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wbi.help.6012.br.ui.doc/topics/rvsdt.html>

To implement the business rule, follow these steps:

- ▶ Expand the *Business Logic* of ClipsAndTacksF2. Open the rules group CheckOrderHandlingPolicyforAutomaticApproval.
- ▶ Select the *InputCriterion* interface and click *Enter Destination*. Select *New Decision Table* and enter a name of AutomaticApprovalFuture2. Click *Next*.
- ▶ Accept the default layout and click *Finish*.
- ▶ The AutomaticApprovalFuture2 rule opens.
- ▶ Add an action rule in the Initialize section (Figure 16-7) and set the action to:
 - Output = Input

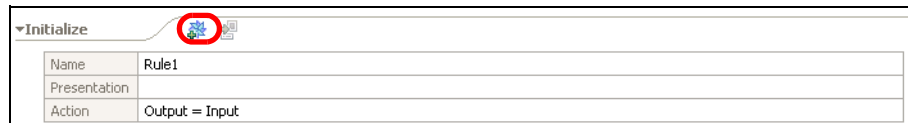


Figure 16-7 Decision table: Initialization rule

Tip: Another option would be: `Output = copyB0(Input)`. With `Output = Input`, changes to the output object are applied to the input object, whereas with a copy we have two business objects.

- ▶ In the *Conditions* section, click *Enter Term*. A selection box appears from which you can select *Input* → *Customer* → *Classification* (Figure 16-8).

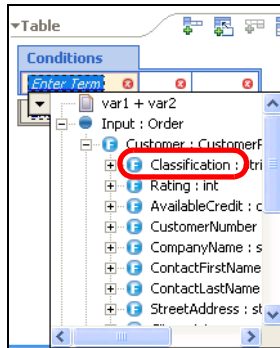


Figure 16-8 Decision table: Select condition

- ▶ Click *Enter Value* to the right of the condition. Select ==, then select *String* and enter GOLD in the text box (Figure 16-9).

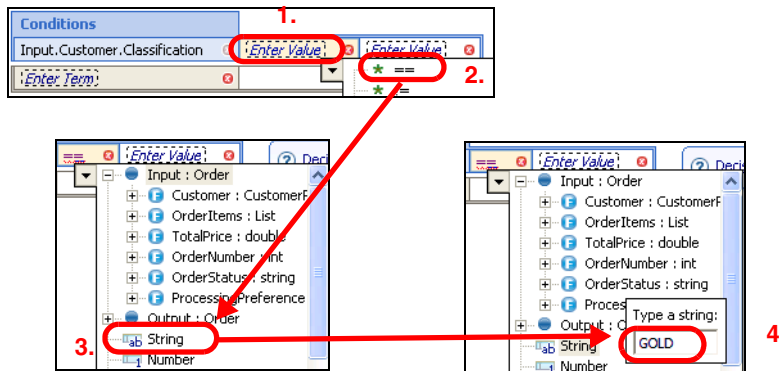



Figure 16-9 Decision table: Enter value for classification

- ▶ Repeat for the next value with a text of SILVER.
- ▶ Click the SILVER cell then right-click and select *Add Condition Otherwise*. Any other value is treated as REGULAR.
- ▶ Right-click in the condition with the text `Input.Customer.Classification` and select *Add Condition*. In the new condition field right-click and select *Change Orientation* (Figure 16-10). You can also click the  icon.

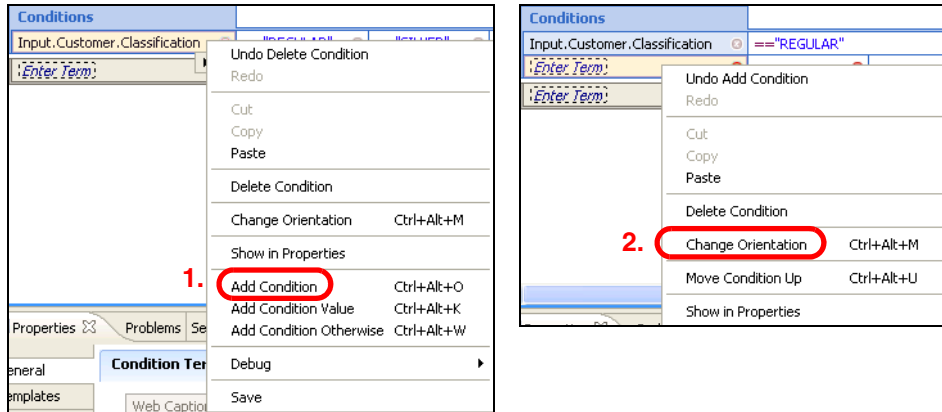


Figure 16-10 Decision table: Add condition and change orientation

- ▶ For *Enter Term* select `Input.TotalPrice`.
- ▶ Enter values for the conditions as shown in Figure 16-11:
 - Click *Enter value*, select `<=` and *Number*. Enter 750.00.
 - Click *Enter value*, select `<=` and *Number*. Enter 1250.00.
 - To add a row, select the cell and *Add Condition Value*.
 - Click *Enter value*, select `<=` and *Number*. Enter 1750.00.
 - Select the bottom cell and *Add Condition Otherwise*.

Conditions			
Input.Customer.Classification	=="GOLD"	=="SILVER"	Otherwise
Input.TotalPrice	<i>Enter Term:</i>	<i>Enter Term:</i>	<i>Enter Term:</i>
<= 750.00			
<= 1250.00			
<= 1750.00			
Otherwise			
Actions			

Figure 16-11 Decision table: Enter values for total price


- ▶ Click *Enter term* and select `Output.ProcessingPreference.automaticApproval`.
- ▶ Enter the values true or false (Figure 16-12).

Conditions			
Input.Customer.Classification	=="GOLD"	=="SILVER"	Otherwise
Input.TotalPrice	<code>Output.ProcessingPreference.automaticApproval</code>	<code>Output.ProcessingPreference.automaticApproval</code>	<code>Output.ProcessingPreference.automaticApproval</code>
<= 750.00	true	true	true
<= 1250.00	true	true	false
<= 1750.00	true	false	false
Otherwise	false	false	false
Actions			

Figure 16-12 Decision table: Completed table (compressed)

Convert the decision table into a template

As with rulesets, we can turn the condition values into templates, which can then be maintained using the Business Rules Manager without redeploying the application.

- ▶ Click into the 750.00 cell, then click the  icon (*Convert Rule or Table Cell Into Template*).
- ▶ Repeat this for the 1250.00 and 1750.00 values.
- ▶ In the Properties view (Figure 16-13) change the name to Total Price Template. Click in the Constraint column and select *Range*. Click *Enter Expression* and select *Inclusive Range* and enter the values 500.00 and 2000.00.
- ▶ Save and close the business rule and the rule group.

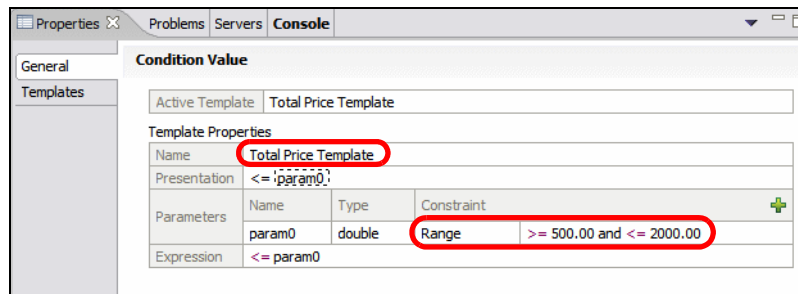


Figure 16-13 Converting the values to a template


Implementing the human tasks

Note: In this section we assume that you run with security. Otherwise the human task configuration is different.

In this application we only use our customized human tasks application described in “Implementing a customized human task application” on page 279.

We do activate the BPC Explorer to potentially look at instances, but we will not configure custom JSPs.

To implement the human tasks follow these steps:

- ▶ Expand the *Business Logic* of ClipsAndTasksF2. Open ReviewOrder.
- ▶ Under *Receiver settings* add *Potential Instance Creator* by clicking .

The default values for the Staff Group Verb in the properties section allows everybody to create this work item (Figure 16-14). Because the process is initiated by a customer from the Web application, this setting is appropriate.

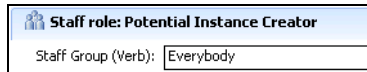


Figure 16-14 Human Tasks: Set properties for Potential Instance Creator

- ▶ Select *Potential Owner* (under Receiver settings) and in the Properties view, set the verb to *Group Members*.
- ▶ Set the following values (Figure 16-15):
 - GroupName: ordermanager
 - IncludeSubgroups: false
 - AlternativeGroupName1: administrator

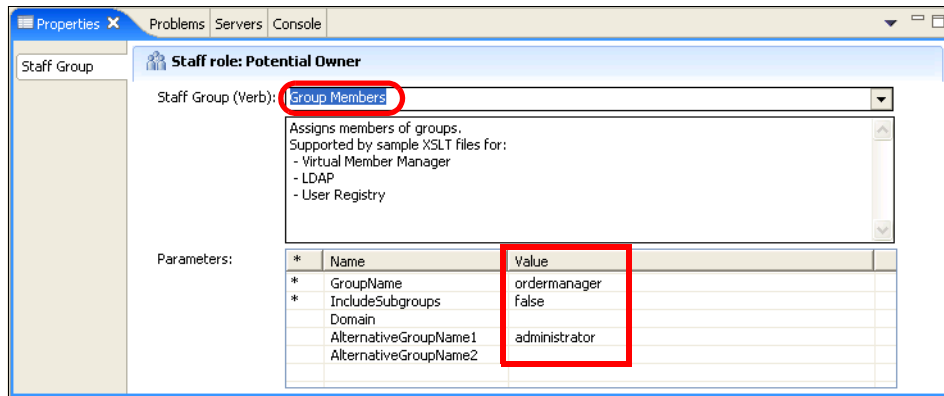



Figure 16-15 Human Tasks: Set properties for Potential Owner

- ▶ Click the *BPC Explorer* icon  under Client settings.
- ▶ Repeat the steps above for *ShipOrderToCustomer*, but substitute ordermanager with shipper as the *Potential Owner*.

Define the staff plug-in for security

In the Properties view Details tab, set the JNDI name for the staff plug-in for both human tasks, depending on your security implementation (Figure 11-24 on page 324):

bpe/staff/userregistryconfiguration	User registry
bpe/staff/sampleldapconfiguration	LDAP

- ▶ Save and close the human task editors.

Adding the Update Order information service for a shipped order

The Update Order information service is the same as the one we used in Order Handling (Future 1). For a detailed discussion of the implementation, refer to “Implementing database update” on page 287.

The code snippet for the SQL statement is in:

```
SG247148\sampcode\widF2\codesnippets\updateOrderSQL.txt
```

Here is a summary of the steps required:

- ▶ Open the Order Handling (Future 2) process.
- ▶ Right-click the canvas and select *Create Information Service Support Types*.
- ▶ Create a new variable and name it `ClipsAndTacksDataSource` of type `tDataSource`. On the Initial Value tab specify the JNDI name `jdbc/clipTack`.
- ▶ Change the Update Order Database implementation type to an information service by selecting the activity and *Change Type* → *Information Service*.
- ▶ Select the Update Order Database component and in the Properties view, Details tab, specify:
 - Kind of information service: *SQL snippet* (click *Yes* in the prompt)
 - Data source: `ClipsAndTacksDataSource`
 - Specify the SQL statement as follows:

```
UPDATE CT.ORDERMAIN SET orderstatus = 'SHIPPED' WHERE ordernumber =
```
 - With the cursor placed immediately after `ordernumber =` select *Parameter* to add a parameter called `orderNumber`, which takes its value from `OrderNumber` in the Order business object.
- ▶ Save the changes.

Implementing the Cancel Order and Send Notification component

This component is largely unchanged compared to the previous implementation, however, we have removed the database access. Refer to “Cancel Order and Send Notification” on page 235 for detailed instructions.

The code snippet for the Java implementation is in:

```
SG247148\sampcode\widF2\codesnippets\cancelOrder.txt
```



Below is a summary of the steps required:

- ▶ Open the Cancel Order and Send Notification component.
- ▶ Add the variables to the class.

- ▶ Update the `CancelOrderandSendNotification_xxxxxxxImpl` constructor.
- ▶ Update the `InputCriterion` method.
- ▶ Add imports for `java.util.List` and `java.math.BigDecimal` to resolve the references to `List` and `BigDecimal`.
- ▶ Save the class.

Adding an information service for a declined order

We have to update the database for a declined order. We use an information service that is invoked after the Cancel Order and Send Notification activity:

- ▶ Open the Order Handling (Future 2) process diagram.
- ▶ Select an Information Service from the Palette  →  and place it under Cancel Order and Send Notification.
- ▶ Name the component Update Order DB Decline, and connect it (click Cancel Order ... and drag to Update Order DB Decline).
- ▶ Configure the information service in the same way as Update Order Database:
 - Type: *SQL snippet*
 - Data source variable: `ClipsAndTacksDataSource`
 - SQL statement with parameter (*Order* → *OrderNumber*):

```
UPDATE CT.ORDERMAIN SET orderstatus = 'DECLINED' WHERE ordernumber =
#orderNumber#
```

- ▶ Connect Cancel Order and Send Notification to the information service (Figure 16-16).

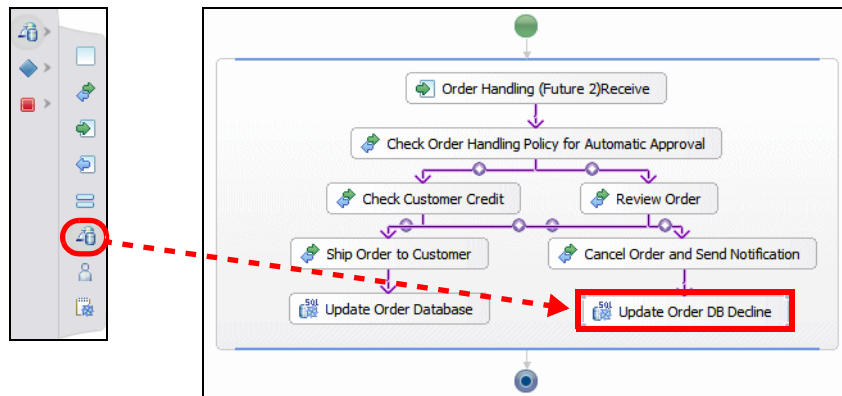


Figure 16-16 Adding an information service for declined orders

Implementing the Check Customer Credit component

The Check Customer Credit component is a process, which we invoke from Order Handling (Future 2). We have exposed the process through an Export as well, so business processes in other projects can reuse the component.

In the assembly diagram, open the Check Customer Credit process.

The Map activity has to initialize the input for the checkCredit activity and pass the original OrderVariable to Update order with Customer Credit. Passing on the OrderVariable is done automatically because the input variable received for the activity is used directly as the input for the Update order with Customer Credit activity.

Therefore, we have to initialize the input for checkCredit. On the first use of a variable, the object is created automatically unless this takes place in a visual snippet. In our case we have to create the input object for checkCredit and then initialize the variable. Follow these steps:

- ▶ Select the checkCredit activity. In the Properties view, we can see that the input is a CustomerBeanVariable.
- ▶ Select the Map activity.
- ▶ In the Properties view, select *Details*, then select *Visual* and click *Yes* when prompted (Figure 16-17).

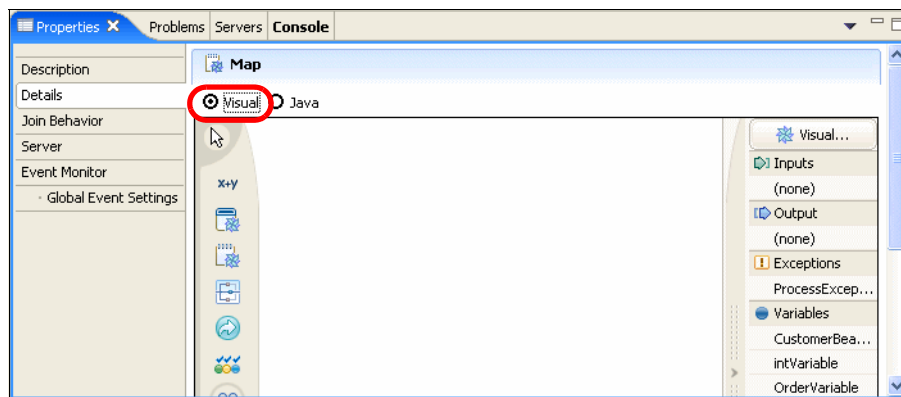



Figure 16-17 Visual Snippet: Changing the implementation to visual

- ▶ In the palette, click  for Standard Visual Snippets.
- ▶ In the pop-up window, select *SCA services* → *create specific BO*. Next select the CustomerBean and click *OK* (Figure 16-18).

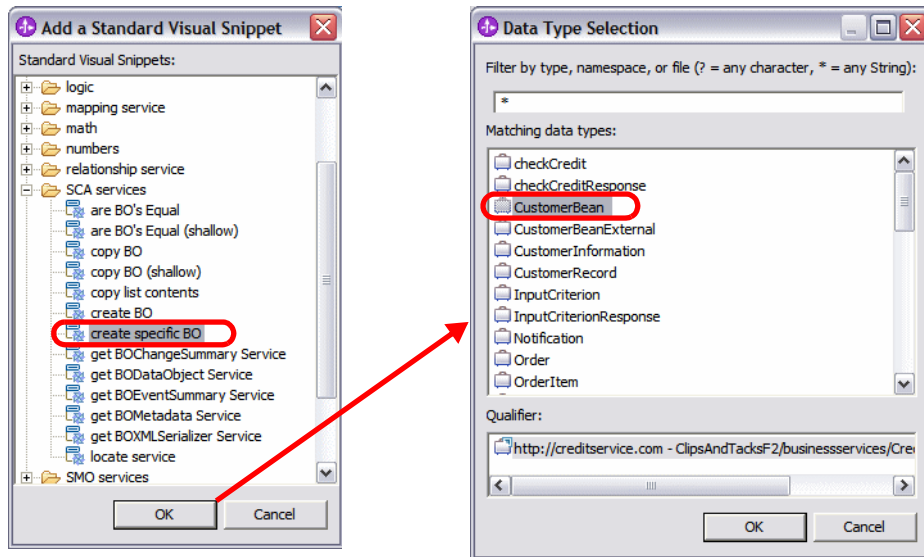


Figure 16-18 Visual Snippet: Create business object

- ▶ The mouse pointer changes shape, which indicates that you are ready to place the object on the canvas. Click anywhere on the canvas and the snippet appears (Figure 16-19).

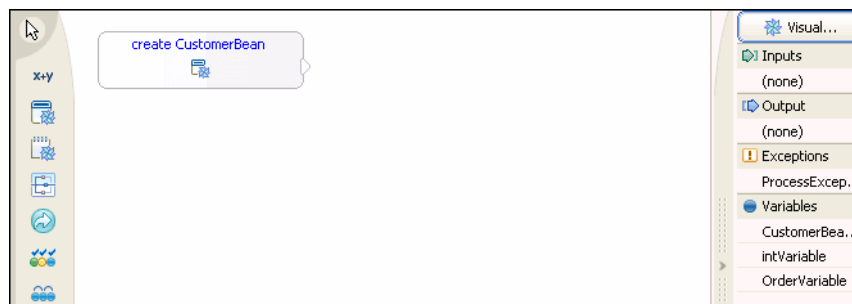


Figure 16-19 Visual Snippet: Placing the CustomerBean on the canvas

- ▶ The business object has now been created, and we have to assign it to the CustomerBeanVariable.
- ▶ In the right-hand Visual Snippet palette, select the CustomerBeanVariable and drag it onto the canvas (Figure 16-20).

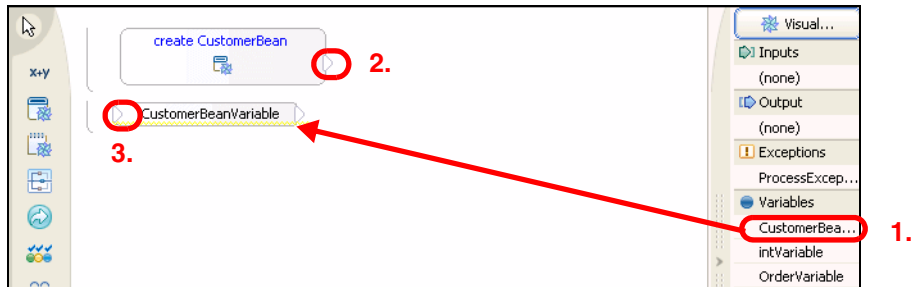


Figure 16-20 Add a CustomerBeanVariable

- ▶ Connect the output from create CustomerBean to the input of CustomerBeanVariable. The diagram rearranges itself automatically.
- ▶ In the right hand pane, select OrderVariable and drag it on to the canvas. Then repeat for CustomerBeanVariable (Figure 16-21).

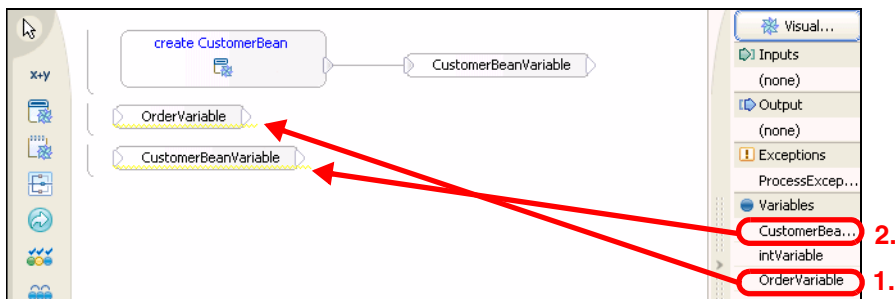


Figure 16-21 Visual Snippet: Dragging variables on to the canvas

- ▶ In the OrderVariable, expand the definition and select OrderVariable.Customer.CustomerNumber (Figure 16-22).

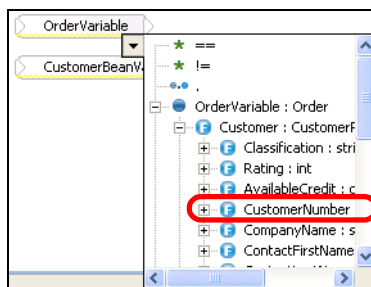


Figure 16-22 Visual Snippet: Expanding variables

- ▶ Expand CustomerBeanVariable to CustomerBeanVariable.customerNumber.

- ▶ Connect `OrderVariable.Customer.CustomerNumber` to `CustomerBeanVariable.customerNumber` (Figure 16-23).

When the Visual Snippet is executed, the customer number is copied from Order to CustomerBean.

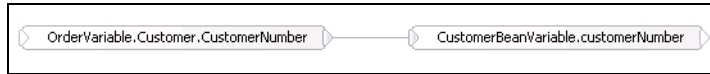


Figure 16-23 Visual Snippet: Connecting variables

- ▶ Drag another `OrderVariable` and `CustomerBeanVariable` on to the canvas and connect `OrderVariable.Customer.CompanyName` to `CustomerBeanVariable.companyName` (Figure 16-24).

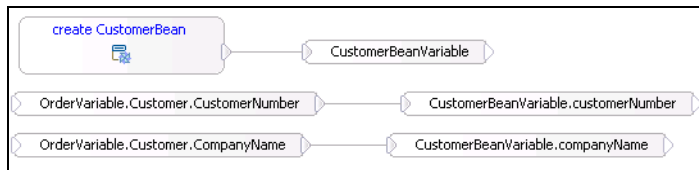


Figure 16-24 Visual Snippet: Completed snippet

- ▶ Save your work and close the BPEL process editor.

Implementing the Update order with Customer Credit component

This component is implemented in Java. The component calculates a new credit rating and available credit based on the old value and the new rating obtained from the Web service. It updates the `Order` business object, but not the database; we add an information service to do this later.

Open the Update order with Customer Credit Java component. One method requires code:

```
public DataObject InputCriterion(int Input2, DataObject Input) {
    //TODO Needs to be implemented.
    return null;
}
```

Complete the method using the sample code:

```
SG247148\sampcode\widF2\cidesnippets\UpdateorderwithCustomerCredit.txt
```

- ▶ Complete the `InputCriterion` method with the code in Example 16-1.
- ▶ Save the changes.

```
public DataObject InputCriterion(int Input2, DataObject Input) {
    double creditAdjustmentFactor = 0.1; //how much to adjust the credit limit
    System.out.println("Update order invoked, new rating: " + Input2);
    if (Input2 == 0) {
        System.out.println("Update order invoked with 0: Web service failure?");
        return Input;
    }
    int newRating = Input2;
    DataObject customer = Input.getDataObject("Customer");
    int customerNumber = customer.getInt("CustomerNumber");
    double availCredit = customer.getDouble("AvailableCredit");
    int oldRating = customer.getInt("Rating");
    System.out.println("Update order old rating/credit: " + oldRating + "/" +
        availCredit + " for customer: " + customerNumber);

    availCredit = availCredit + (newRating - oldRating) *
        creditAdjustmentFactor;
    customer.setInt("Rating", newRating);
    customer.setDouble("AvailableCredit", availCredit);
    System.out.println("Update order new rating/credit: " + newRating + "/" +
        availCredit);

    return Input;
}
```

Updating the customer information in the database

After updating the customer credit information in the order, we also have to update the CLIPTACK database with the latest information. We use an information service:

- ▶ Open the Check Customer Credit process (Figure 16-25).
- ▶ Add more space at the bottom and move the Check Customer CreditReply down.
- ▶ Add an information service activity between Update order with Customer Credit and Check Customer CreditReply.
- ▶ Name the information service Update Customer in Database.
- ▶ Connect the three activities.
- ▶ Change the information service to an SQL snippet, then:
 - Define the ClipsAndTacksDataSource variable of type tDataSource.
 - Initialize the variable with jdbc/cliptack.
 - Use the ClipsAndTacksDataSource variable in the information service.

- Add the SQL statement with three parameters:

```
UPDATE CT.CUSTOMER SET rating = #customerRating# , creditlimit =
      #customerCredit# WHERE customernumber = #customerNumber#
```

The three parameter values come from the *OrderVariable* → *Customer*: Rating, AvailableCredit, and CustomerNumber.

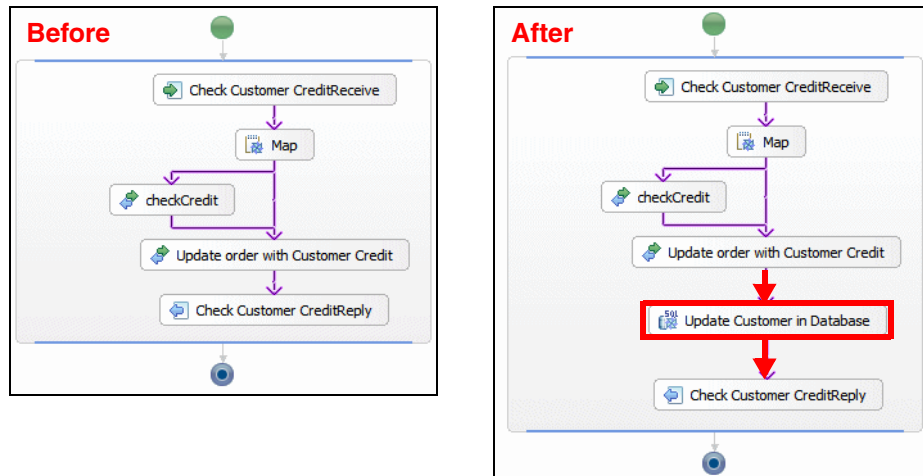


Figure 16-25 Adding an information service to update the customer

Importing the custom human task application

The custom human task application `ClipsAndTacksFxHumanCustomEAR` does not change significantly for Future2. The customer classification is now retrieved from the Order business object and displayed in the JSPs, and the database update is deferred to an information service.

The changes in the human task application from Future 1 to Future 2 are as follows:

- ▶ The `BusinessItems.xsd` file includes the customer classification and uses a different namespace (`http://BusinessItemsF2`) to distinguish the objects. We copied the file into the application as `BusinessItems2.xsd`
- ▶ The `TaskQuery` class handles the customer classification.
- ▶ The update of the `CLIPTACK` database with new customer rating and available credit is not performed. The database is updated using an information service in the Check Customer Credit process.

We already included the necessary code in the human task application provided for Future 1, therefore no action is required.

Importing the Web front-end application

Remember that, in Order Handling (Future 1) we invoked the process through a stand-alone reference. With that approach we have to attach the Web project to the enterprise application of the process. Consequently, if there are any changes to the Web application, we have to redeploy the enterprise application for the process. This is not desirable. However, if we invoke the business process through a Web service call, this dependency does not exist, and we have much better control over the deployment of individual components of the solution.

To invoke the business process using a customer order, we use a Web front-end application as in Future 1, but we explore two options:

1. Using an integrated Web front-end in the same way as for Future 1
2. Using a Web front-end that invokes the business process as a Web service

Using a Web front-end with an SCA interface

This solution is identical to the Future 1 application. The only minor change is that the customer classification (the new data item) is passed in the business object. Follow the instructions in “Implementing a Web front-end” on page 247:

- ▶ The Java module `Cl ipsAndTacksF1Database` is already imported. Add it to the `Cl ipsAndTacksF2App` enterprise application using the Dependencies editor.
- ▶ Import the `Cl ipsAndTacksF2Invoke Web` application. Add the Java module to the build path. Add the Web module to the enterprise application using the Dependencies editor. The Web front-end is available in:

```
SG247148\sampcode\widF2\webfrontSCA
```

Note that the context root is `Cl ipsAndTacksF2Invoke` to be different from Future 1.

Using a Web front-end with a Web service interface

A front-end application that invokes the business process as a Web service has these advantages:

- ▶ Because the Future 1 application is still running in the real server in parallel to the new Future 2 application, we have to use a different context root.
- ▶ We also know that the integrated Web application must be a part of the enterprise application of the business process.
- ▶ A Web service front-end is a stand-alone application that can be replaced at any time.
- ▶ The invocation as a Web service requires the generation of Web service client code based on the Web service interface that we generate for the export of the process.

Restriction: The implementation of the Web front-end as a Web service works properly in the test environment under Integration Developer. However, the application fails in the stand-alone Process Server. Errors are reported when the business process application is deployed to the server and started.

This error has been fixed with APAR - PK35442, scheduled for cumulative fix 6.0.2.19:

<http://www-1.ibm.com/support/docview.wss?uid=swg24015274>

To implement the Web front-end using a Web service call, follow the instructions in this section.

We have to activate the Web service development capability in the Workbench:

- ▶ Select *Window* → *Preferences*. In the Preferences dialog, expand *Workbench* and select *Capabilities*.
- ▶ Select *Web Services Developer* and click *OK*.

Import the front-end application

To import the skeleton front-end application, follow these steps:

- ▶ In the Web perspective, select *Dynamic Web Projects* and *Import* → *WAR file*.
- ▶ Click *Browse* to locate the WAR file:
SG247148\sampcode\widF2\webfrontWS\C1ipsAndTacksF2InvokeWS.war
- ▶ Keep all the default values. An enterprise application C1ipsAndTacksF2InvokeWSEAR is created as well.
- ▶ Click *Finish*.

Generate Web service binding for Order Handling Future 2

We will invoke the Order Handling (Future 2) process through a Web service call from the front-end Web application.

Therefore, we have to generate a Web service binding for the Order Handling (Future 2) Export in the assembly diagram.

Perform these steps:

- ▶ Select the Order Handling (Future 2) Export and *Generate Binding* → *Web Service Binding*.
- ▶ Select *soap/http* as the transport (Figure 16-26).

The Web service binding is now populated. Save the assembly diagram.

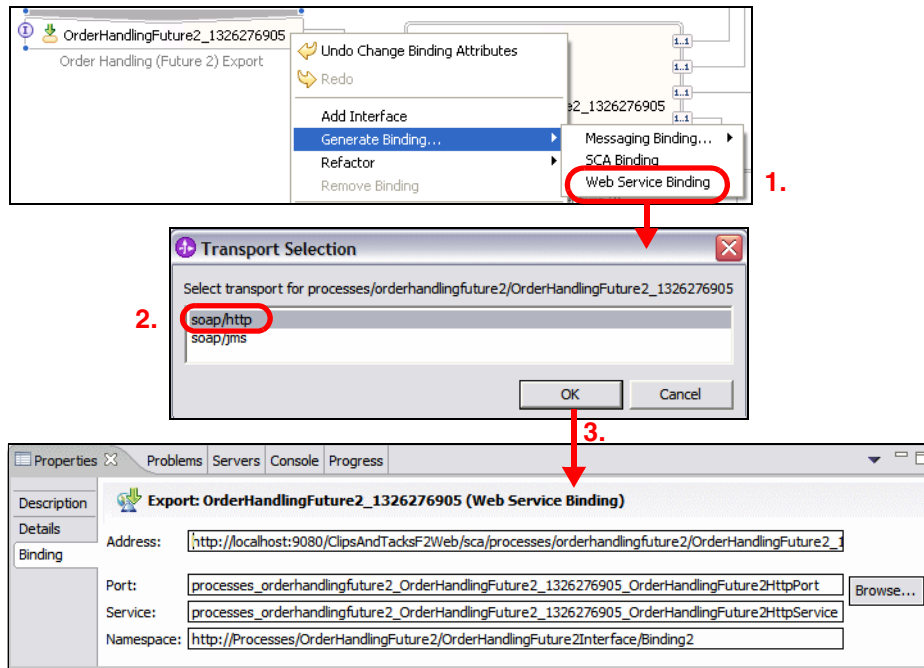


Figure 16-26 OrderHandlingFuture2: Generate Web service binding for export

Locate the service WSDL file in the `processes.orderhandlingfuture2` package of the `ClipsAndTacksF2` project. Open the file and notice that it imports the interface WSDL file, which imports the XSD file of the business items.

```
processes.orderhandlingfuture2
  OrderHandlingFuture2_1326276905_OrderHandlingFuture2Http_Service.wsdl
  imports OrderHandlingFuture2Interface.wsdl
  imports ../../businessitems/Businessitems.xsd
```

These files are now being used as a base for generating the client code.

Generate the Web service client code

From the WSDL and XSD files, we generate the client code:

- ▶ Select the service WSDL file and *Web Services* → *Generate Client*:
`OrderHandlingFuture2_1326276905_OrderHandlingFuture2Http_Service.wsdl`
- ▶ Clear *Test the Web service* and *Monitor the Web service* (if they are selected).
- ▶ Go through the panels. In the Client Environment Configuration panel, click *Edit* and select *IBM WebSphere* for the Web service runtime. Select *Web* for client type and select the imported Web project (Figure 16-27).

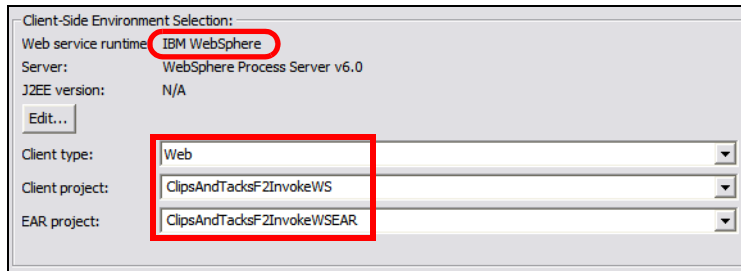


Figure 16-27 Web Service wizard client selection

- ▶ Click *Finish* on the last panel.

Next, let us explore the generated code in the ClipsAndTacksF2InvokeWS project (Figure 16-28).

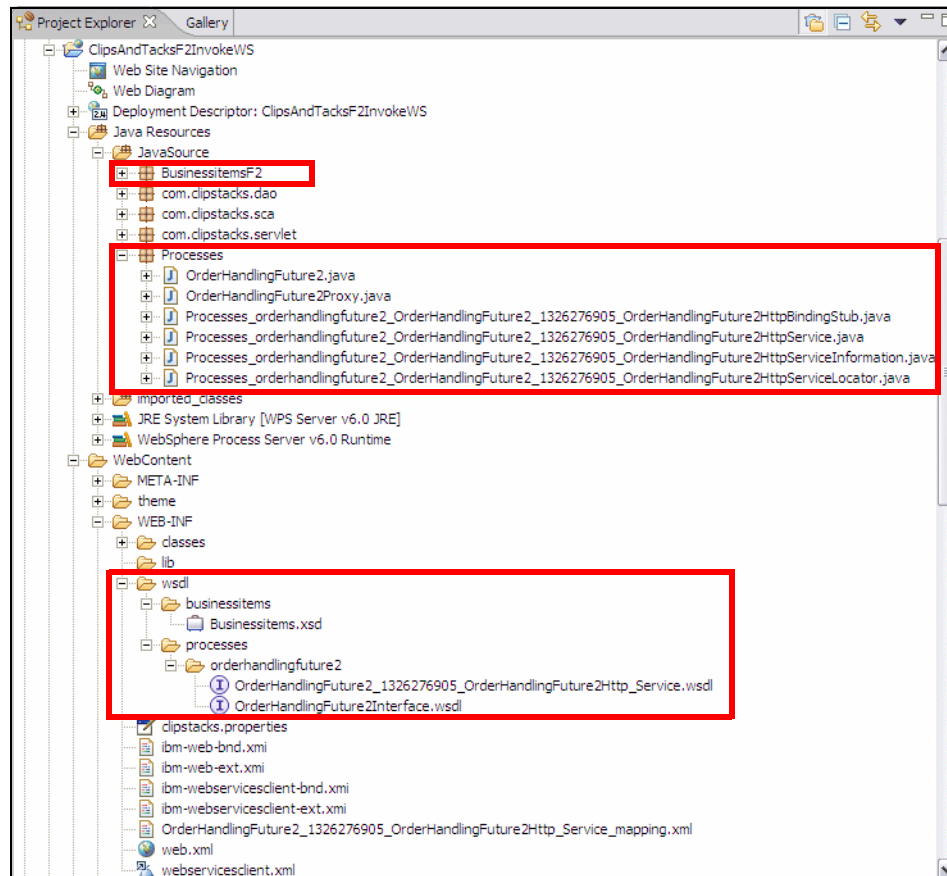


Figure 16-28 Generated Web service client

- ▶ The generated data transfer objects (DTO) and helper classes are located in the `BusinessItemsF2` package under *Java Resources* → *JavaSource*.
- ▶ The generated Web service interface, proxy, and helper classes are located in the `Processes` package.
- ▶ The XSD file and the WSDL files are copied into the Web project (`WEB-INF`).

Completing the front-end code

We have to update some of the code to make it executable.

- ▶ Open `Order.java` and update the generated code for the `getTotalPrice` method (the code is in `order.txt` in the `sampcode\widF2\webfrontWS` folder):

```
public double getTotalPrice() {
    totalPrice = 0.00;
    if (orderItems == null) return totalPrice;
    for (int i=0; i < orderItems.length; i++) {
        totalPrice = totalPrice +
            orderItems[i].getPrice() * orderItems[i].getQuantity();
    }
    return totalPrice;
}
```

- ▶ The `CustomerInformation` and `CustomerRecord` beans are somewhat different to the `Customer` DTO in `Future 1` and we had to change the JSPs for a new customer to use the new method names (`getPostalCode`, for example).
- ▶ In the `com.clipstacks.dao` package, open the `ClipsTacksDAOWS.java` file and replace the code with the code in `ClipsTacksDAOWS.java` in the `sampcode\widF2\webfrontWS` folder.

This data access object uses the generated DTOs to access the `CLIPTACK` database. This is basically the same code as in the `Future 1` application.

- ▶ In the `com.clipstacks.sca` package, open the `InvokeOrderHandlingWS.java` file and replace the code with the code in `InvokeOrderHandlingWS.java` in the `sampcode\widF2\webfrontWS` folder.

The `InvokeOrderHandlingWS` code uses the generated Web service proxy (`OrderHandlingFuture2Proxy`) to invoke the business process as a Web service (abbreviated):

```
public boolean execute(Order order) {
    OrderHandlingFuture2Proxy proxy = new OrderHandlingFuture2Proxy();
    proxy.setInputCriterion(order);
}
```

This code is much simpler than the code we had to write to invoke the business process using the SCA interface. We can pass the simple `Order` DTO instead of having to create the complex `Order` business object.

Deploying and testing the Future 2 application

The implementation is now complete and we can deploy and test the business process in the built-in Process Server:

- ▶ Start the Process Server and wait until it is ready.
- ▶ Select the server in the Servers view and *Add and remove projects*:
 - Select these projects:
 - ClipsAndTacksF2App
 - ClipsAndTacksF2InvokeWSEAR
 - ClipsAndTacksFxHumanCustomEAR
 - CreditServiceEAR
 - Click *Finish*.

Wait until all the applications are started in the server.

Note: If you generated the Web service binding for the export, then errors might be reported about not finding the XSD file. This happens if you specify to run with resources in the server (server configuration). The application still works, because we can invoke the process using the integrated Web front-end. Refer to the **Restriction** on page 529 for information about the APAR Fix.

Testing the application

Refer to “Using the Web front-end” on page 263 for instructions on how to submit an order. Then refer to “Using the human task application” on page 284 for instructions on how to handle the human tasks.

The integrated Web front-end is invoked by this URL (note that you have to use the changed context root for the Web application):

```
http://localhost:9080/ClipsAndTacksF2Invoke/
```

The human task application is invoked through this URL (be sure to select *Future 2* in the pull-down menu):

```
http://localhost:9080/ClipsAndTacksFxHumanCustomWeb/
```

You can also submit orders using the Web service front-end:

```
http://localhost:9080/ClipsAndTacksF2InvokeWS/
```

Changing the port numbers

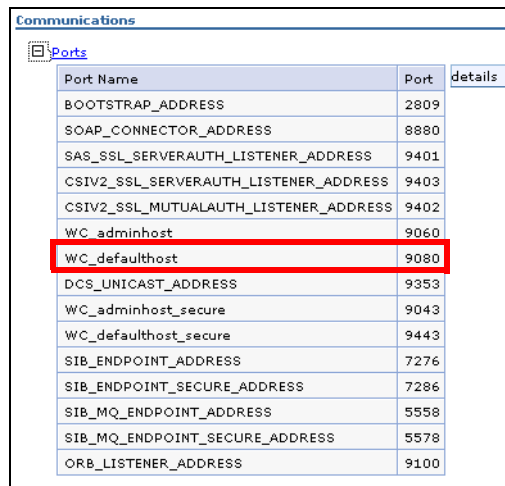
If you have more than one server defined, your port numbers may be different. The first server you define is assigned port 9080 for HTTP traffic, the second server 9081, and so forth. This is to avoid port number conflicts when running more than one server at the same time.

Note: You only have to change the port numbers manually if you *imported* the solution from the sample code and your server is *not* running on port 9080.

If you built the application yourself or if your server is using port 9080, you do not have to change anything.

You can determine the port numbers of the server in the administrative console:

- ▶ Select *Servers* → *Application servers* → *server1*.
- ▶ In the Communications section, expand *Ports*.
- ▶ Verify the port setting for *WC_defaulthost* (Figure 16-29):



Port Name	Port	details
BOOTSTRAP_ADDRESS	2809	
SOAP_CONNECTOR_ADDRESS	8880	
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9401	
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9403	
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9402	
WC_adminhost	9060	
WC_defaulthost	9080	
DCS_UNICAST_ADDRESS	9353	
WC_adminhost_secure	9043	
WC_defaulthost_secure	9443	
SIB_ENDPOINT_ADDRESS	7276	
SIB_ENDPOINT_SECURE_ADDRESS	7286	
SIB_MQ_ENDPOINT_ADDRESS	5558	
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5578	
ORB_LISTENER_ADDRESS	9100	

Figure 16-29 Changing ports: Determining the server port number

If the port setting is not 9080, follow these steps:

- ▶ Change the port setting in the `checkCredit` Web service binding (Figure 16-30):
 - In the assembly diagram, select the `checkCredit` import.
 - In the Properties view, select the *Binding* tab.
 - Edit the port number in the Address field.

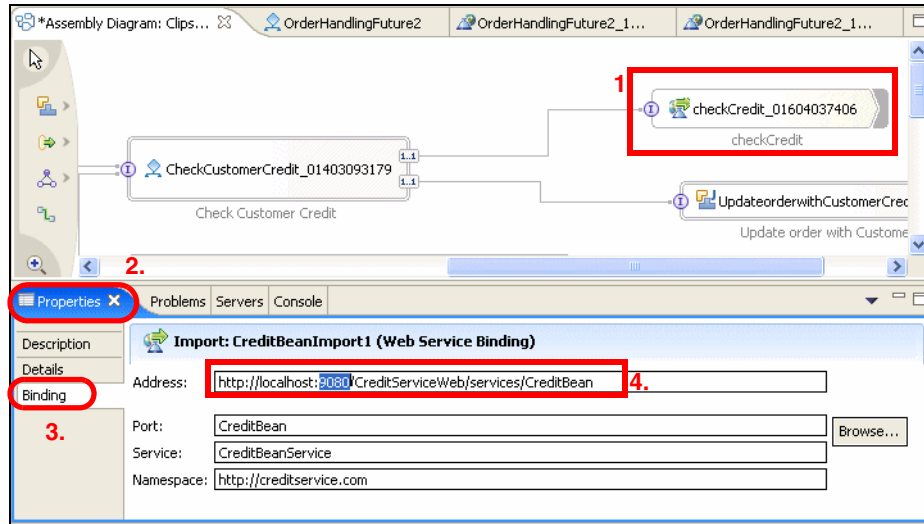


Figure 16-30 Changing ports: checkCredit Web service bindings

- ▶ Change the port for the Order Handling (Future 2) Export:
 - In the Web or J2EE perspective, navigate to *Other Projects* → *ClipsAndTacksF2* → *processes.orderhandlingfuture2*.
 - Select the *OrderHandlingFuture2_...Future2Http_Service.wsdl* and *Open With* → *WSDL Editor*.
 - Change the soap address location:


```
From: <soap:address location="http://localhost:9080/ClipsAndTacks...
To: <soap:address location="http://localhost:xxx/ClipsAndTacks...
```
- ▶ Change the corresponding port in the Web front-end application:
 - Expand *Dynamic Web Projects* → *ClipsAndTacksF2InvokeWS* → *WebContent* → *WEB-INF* → *wsdl*.
 - Select the *OrderHandlingFuture2_...Future2Http_Service.wsdl* and *Open With* → *WSDL Editor*. Change the soap:address location:


```
From: <soap:address location="http://localhost:9080/ClipsAndTacks...
To: <soap:address location="http://localhost:xxx/ClipsAndTacks...
```
 - Open the *Processes.Processes_orderhandling...HttpServiceLocator* Java class and change the port in the final string with the address:


```
processes_order..._address = "http://localhost:xxx/ClipsAndTacks...
```

Now you are ready to run the application.

Implementing a new Web service

Our process invokes a Web service to obtain a credit rating, which is used for further processing. Now imagine the situation where the interface of the Web service that we invoke changes.

One reason for this could be that our development environment used an internal Web service and we will go to production using an external Web service. Another reason could be that the company providing credit ratings implements changes to their service, or perhaps we are substituting the credit rating company with a more cost efficient competitor.

The end result is the same: We have to invoke a new Web service with a different set of operations and parameters.

How will this affect our business process?

Fortunately WebSphere Process Server provides us with capabilities to map between a service consumer expecting one format and a service provider delivering another format. In our case the service consumer is the Check Customer Credit process and the service provider is the `checkCredit` import.

The change is implemented using an interface map, which maps from one operation to another. Parameters are transformed using business object maps, which can perform complex transformations between different business objects.

As a consequence, the impact of the change is minimal.

Importing the new Web service

The new Web service has been provided for you. It is very similar to the old Web service—in fact, the only change is in the naming, but it is sufficient to illustrate the principle.

Follow these steps:

- ▶ Select *File* → *Import*.
- ▶ Select *EAR file* as the file type and click *Next*.
- ▶ Click *Browse* and locate the file:
SG247148\sampcode\widF2\webservice\CreditServiceExternalEAR.ear.
- ▶ Click *Finish*.

Copy the Web service WSDL to the business process

To use the Web service in our business integration project, we have to copy the imported WSDL file to our library project:

- ▶ In the J2EE perspective, copy the file `CreditBeanExternal.wsdl`
 - From: Dynamic Web Projects → `CreditServiceExternalWeb` → `WebContent` → `WEB-INF` → `wsdl`
 - To: Other Projects → `ClipsAndTacksF2` → `businessservices`
- ▶ The `CreditBeanExternal` port appears under `ClipsAndTacksF2` → `Web Service Ports`.

Updating the business process

The business process has to be wired to the new Web service in the assembly diagram.

To accomplish this, follow these steps:

- ▶ In the business integration view, open the assembly diagram.
- ▶ Delete the import (`checkCredit`) for the old Web service (Figure 16-31).

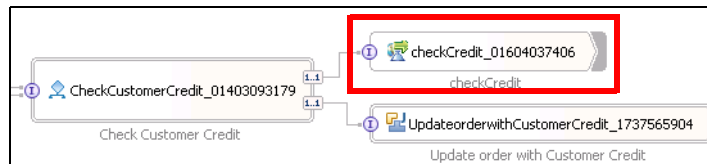


Figure 16-31 Delete import for old Web service

- ▶ Drag `ClipsAndTacksF2` → `Web Service Ports` → `CreditBeanExternal` to the canvas. Select `Import with Web Service Binding` (Figure 16-32).

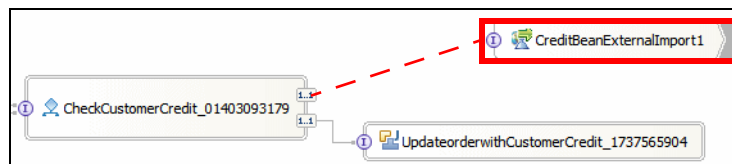


Figure 16-32 Add new Web service

- ▶ Connect the available reference from `CheckCustomerCredit` to the interface of `CreditBeanExternalImport1`.
- ▶ In the resulting prompt that the interfaces do not match, specify a name of `CheckCreditToExternalServiceMap` and click `OK` (Figure 16-33).

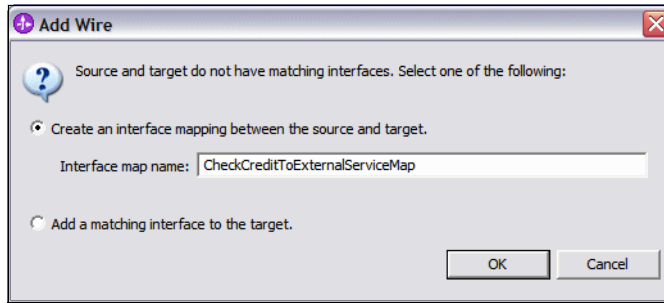


Figure 16-33 Create an interface map

- ▶ The interface map is placed and wired between the process and the Web service (Figure 16-34).

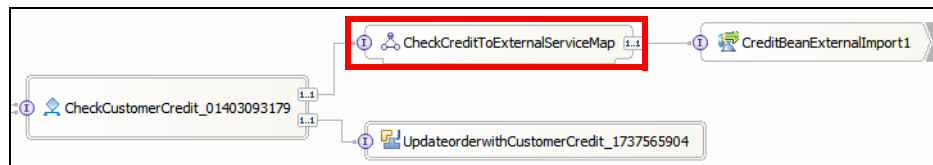


Figure 16-34 Interface map between process and Web service

Creating the interface map

Find and open the `CheckCreditToExternalServiceMap` interface map in the Business Integration view under *ClipsAndTacksF2* → *Mapping* → *Interface Maps*.

- ▶ Connect the `checkCredit` operation of the `CreditBean` interface with the `checkCredit` operation on the `CreditBeanExternal` interface (Figure 16-35).

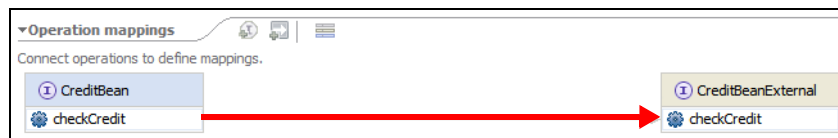


Figure 16-35 Editing the interface map

- ▶ Click the connection to view the parameter mappings.
- ▶ Connect the customer parameter in the `CustomerBean` to the customer parameter in the `CustomerBeanExternal` to map the input parameter.
- ▶ Connect the `checkCreditReturn` in the `CustomerBeanExternal` to the `checkCreditReturn` in the `CustomerBean` to map the return parameter, from **right to left** (Figure 16-36).

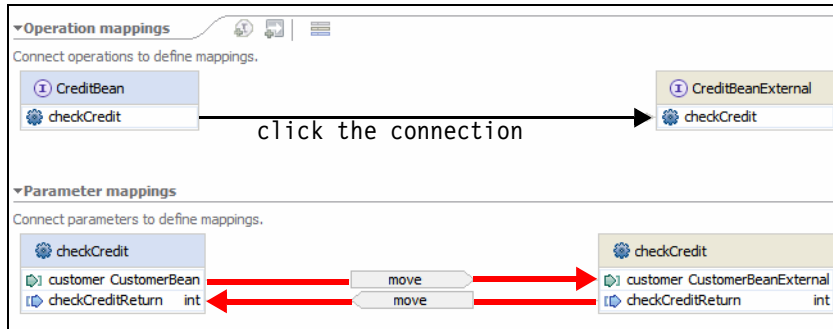


Figure 16-36 Create the mapping

- ▶ Select the move operation on the customer to customer connection. In the Properties view, change the Parameter Mapping Type from move to map (Figure 16-37).

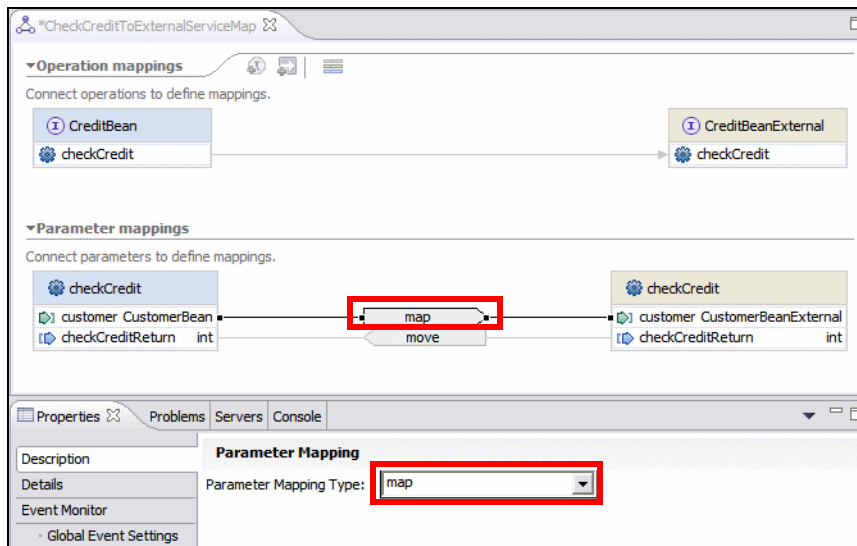


Figure 16-37 Defining the interface map

Creating the business object map

We have to map the CustomerBean to the CustomerBeanExternal:

- ▶ Select the *Details* tab.
- ▶ Click *New* to create a business object map:
 - Enter a name of CustomerBeanToCustomerBeanExternalMap.
 - Click *Next*.
 - Leave the default values and click *Finish*.

- ▶ The CustomerBeanToCustomerBeanExternalMap map opens.
- ▶ Connect customerNumber in CustomerBean to customerNumber in CustomerBeanExternal. Leave the default Move transformation type.
- ▶ Connect companyName in CustomerBean to companyName in CustomerBeanExternal. Leave the default Move transformation type (Figure 16-38).

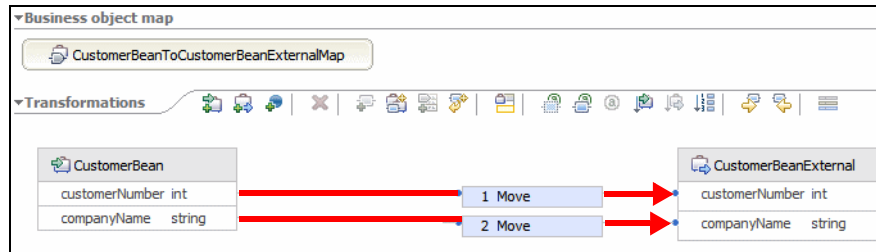


Figure 16-38 Defining the business object map

- ▶ Save and close the business object map.
- ▶ Save and close the interface map.
- ▶ Save the changes in the assembly diagram.

The maps are located in the Mapping section in the Business Integration view (Figure 16-39).

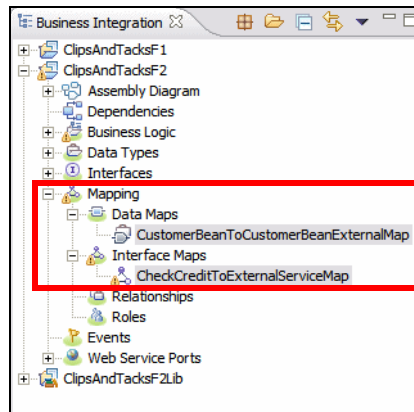
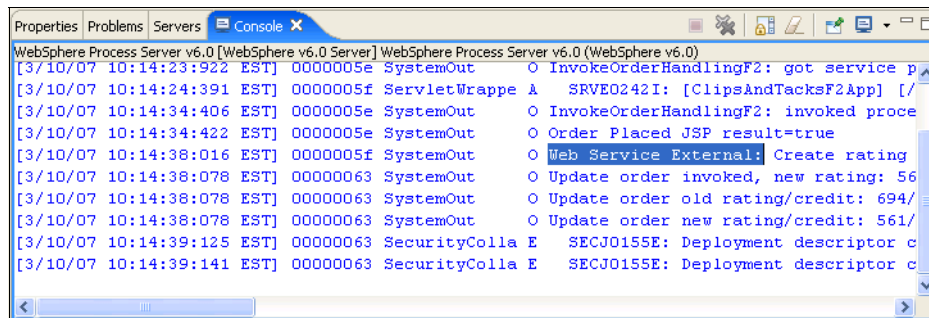


Figure 16-39 Location of the interface map and the and business object map

Deploying and testing the application with the new Web service

To execute the modified application, we have to deploy the new Web service to the server.

- ▶ Add the CreditServiceExternalEAR project to the server using *Add and remove projects*.
- ▶ Test the application by running an order that automatically approved.
- ▶ Observe the output in the console to verify that the new Web service is in fact invoked (Figure 16-40):



```
WebSphere Process Server v6.0 [WebSphere v6.0 Server] WebSphere Process Server v6.0 (WebSphere v6.0)
[3/10/07 10:14:23:922 EST] 0000005e SystemOut      O InvokeOrderHandlingF2: got service p
[3/10/07 10:14:24:391 EST] 0000005f ServletWrapp A   SRVE0242I: [ClipsAndTacksF2App] [/
[3/10/07 10:14:34:406 EST] 0000005e SystemOut      O InvokeOrderHandlingF2: invoked proce
[3/10/07 10:14:34:422 EST] 0000005e SystemOut      O Order Placed JSP result=true
[3/10/07 10:14:38:016 EST] 0000005f SystemOut      O Web Service External: Create rating
[3/10/07 10:14:38:078 EST] 00000063 SystemOut      O Update order invoked, new rating: 56
[3/10/07 10:14:38:078 EST] 00000063 SystemOut      O Update order old rating/credit: 694/
[3/10/07 10:14:38:078 EST] 00000063 SystemOut      O Update order new rating/credit: 561/
[3/10/07 10:14:39:125 EST] 00000063 SecurityColla E   SECJ0155E: Deployment descriptor c
[3/10/07 10:14:39:141 EST] 00000063 SecurityColla E   SECJ0155E: Deployment descriptor c
```

Figure 16-40 Invoking the new Web service

Implementing a fault handler

A fault handler can catch execution failures, for example, if the external Web service fails. Without a fault handler, the process instance is terminated and the end user may not be aware of the problem.

We can implement a fault handler to catch the failure of the Web service invocation and let the process continue without a new customer rating. To implement the fault handler, open the CheckCustomerCredit process:

- ▶ Select the **checkCredit** invocation and *Add Fault Handler*.
- ▶ A fault handler box with a catch block appears next to the activity (Figure 16-41).
- ▶ Right-click in the fault handler box and select *Add Catch All*.
- ▶ Select the *catch* block and *Delete*.
- ▶ Select the *Catch All* block and *Add* → *Snippet*. Overtyping the name of the snippet with `checkCreditFault`.

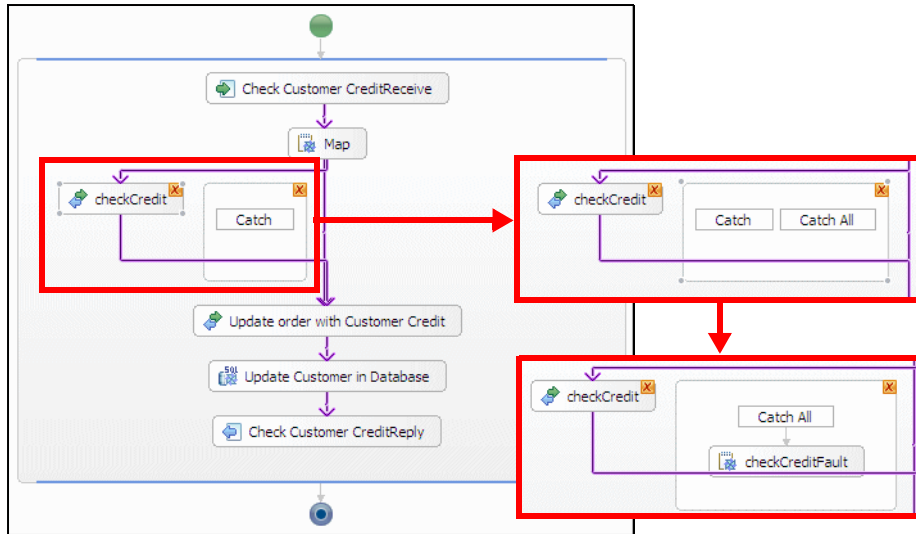


Figure 16-41 Adding a fault handler to the Web service call

- ▶ Select the snippet and in the Properties view, Details tab, add a visual snippet to write a message to the console log (Figure 16-42).

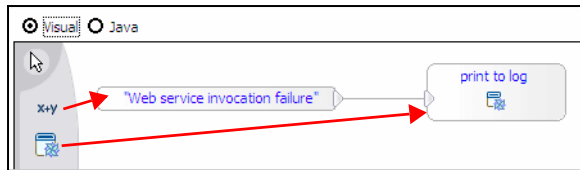


Figure 16-42 Visual snippet with message to the console

- ▶ Save the process.

If the Web service fails, the fault handler is invoked and writes the console message. The process continues with a zero result for the new rating. The next activity, Update order with Customer Credit, must handle the case of a zero rating. Our code already includes the logic to bypass updating of the customer credit with the new rating.

You can test the behavior by removing the Web service application (CreditServiceEAR or CreditServiceExternalEAR) from the server and submitting a small order that runs through the Web service invocation.

Installing and running the application in Process Server

To deploy the application to a real process server, follow the instructions in Chapter 13, “Deploying and running the application in Process Server” on page 401, except that we are running the Future 2 application.

Define human task staff plug-in for security

If the Process Server runs with another security implementation than the test environment, change the staff plug-in configuration for the two human tasks.

In the Properties view Details tab, set the JNDI name for the staff plug-in for both human tasks, depending on your security implementation (Figure 11-24 on page 324):

bpe/staff/userregistryconfiguration	User registry
bpe/staff/sampleldapconfiguration	LDAP

Exporting the Future 2 application

You have to export all the enterprise applications involved:

- | | |
|---------------------------------|-------------------------|
| ▶ ClipsAndTacksF2 | Business process |
| ▶ ClipsAndTacksF2InvokeWSEAR | Invoke as a Web service |
| ▶ ClipsAndTacksFxHumanCustomEAR | Human task application |
| ▶ CreditServiceEAR | Old Web service |
| ▶ CreditServiceExternalEAR | New Web service |

Installing the application

Install the enterprise applications as described in “Installing the application in Process Server” on page 405.

Running the application

Run the application using the URL to submit orders:

<http://localhost:9080/ClipsAndTacksF2Invoke/>

Handle the human tasks using the custom human task application:

<http://localhost:9080/ClipsAndTacksFxHumanCustomWeb/>

Testing the business rule

To test the business rule using the customer classification, remember which customers have which classification:

Customer	Classification
12345	GOLD
11111	SILVER
22222	REGULAR
33333	GOLD
44444	SILVER

- ▶ Automatically approved orders:
 - For a REGULAR customer up to \$750
 - For a SILVER customer up to \$1250
 - For a GOLD customer up to \$1750
- ▶ Such orders might still be declined by the customer credit check. For example, customer 22222 has a low credit limit and an order of \$600 is automatically approved, but declined by the credit check.

Summary

In this chapter we described how to implement the Order Handling (Future 2) process in Integration Developer. We used different techniques as compared to the Future 1 process to illustrate more functions of Integration Developer and Process Server.

The business process is now ready for measurements using the Business Monitor.



Measuring the Future 2 process

In this chapter we implement the WebSphere Business Monitor for the improved Order Handling (Future 2) process.

Then we measure the business process application and verify that we can meet the key performance indicators.

To develop the monitor model in WebSphere Integration Developer, we used a slightly different technique to develop the measures, but the basic concepts are the same as in Chapter 12, “Developing and testing the business measures with the Monitor Toolkit” on page 327.

To deploy and measure the Order Handling (Future 2) process, we used the same technique as described in Chapter 14, “Deploying the monitor model and measuring the Future 1 process” on page 415.

Creating the monitor model for Future 2

We create the monitor model very similar to the model for Future 1.

Preparing the project and the model

To create the project and the model for the business measures, we have to execute the following steps:

- ▶ Generate CEI events for BEPL elements for the Order Handling (Future 2) process. We ignore the Check Customer Credit process because none of our measures that we require are in this process. Refer to “Generating CEI events for BPEL elements” on page 334 for instructions.
- ▶ Generate monitoring events for the Order Handling (Future 2) process. Refer to “Generating monitoring events” on page 337.
- ▶ Generate the monitor model for the Order Handling (Future 2) process. Name the project and the model `C1ipsAndTacksF2BMP`. Refer to “Generating the monitor model” on page 338.
- ▶ Generate monitoring events for the two human tasks. Refer to “Create the human tasks events for Review and Ship Order” on page 372.
- ▶ Generate the monitor models for the two human tasks. Name the models `C1ipsAndTacksF2ReviewBMP` and `C1ipsAndTacksF2ShipBMP`.

Merging the models

For Future 2, we merge the monitor models immediately. Follow the instructions in “Merging the monitor models” on page 373 to copy the monitoring context, the cube, and the events from the human task models to the main model.

In addition, we merge the visual model exported from the Modeler into the new monitor model.

Simplified approach

In this chapter we define the measures in a simpler way than for Future 1:

- ▶ We create an Order Count measure without creating a metric.
- ▶ We reduce the number of triggers.
- ▶ All the metrics are created in the parent monitoring context (`OrderHandlingFuture2`). Creating them in the parent monitoring context gives us the option of providing the dimension by Order Country and City.

- ▶ One additional dimension is created for customer classification so that we can run dimensional analysis against the classification of the customer.
- ▶ Two duration measures (Review Order Process Time and Ship Order to Customer Process Time) and one decision percentage (Check Order Handling Policy For Automatic Approval %) are created to measure the effectiveness of the improved process. In this chapter we are not implementing the round-trip to the Modeler, but we show how to measure two durations and a decision percentage.

Creating the metrics and the measures

In this section we describe how to create all the measures required. We do not describe the detailed steps for each measure. We show the table that outlines the steps to create the measures.

Order Count

To create the Order Count measure, we do not creating any metrics; we use the ProcessInstanceIDFact already available (Table 17-1).

Table 17-1 Order Count measure details

Activity	Description / Location	Name / Value
Add measure	Data Mart Model → Measures: OrderHandlingFuture2 Cube	Name: Order Count Source: ProcessInstanceIDFact Aggregation Function: Count

Shipped Order Count

Create the Shipped Order Count in monitoring context OrderHandlingFuture2 (Table 17-2).

Table 17-2 Shipped Order Count measure details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: OrderHandlingFuture2	Name: Order_Shipped
Associate event to trigger	Trigger Source: ShipOrdertoCustomer_Input Criterion	Select: ShipOrdertoCustomer_InputCriterion EXIT
Create counter	Counter Location: OrderHandlingFuture2	Name: Shipped Order Count

Activity	Description / Location	Name / Value
Associate trigger to counter	Counter Controls: OrderHandlingFuture2	Trigger: Order_Shipped Resulting action: Add One
Add measures	Data Mart Model → Measures: OrderHandlingFuture2 Cube	Name: Shipped Order Count Source: Shipped Order Count Fact Aggregation Function: Sum

Declined Order Count

Create the Declined Order Count (Table 17-3).

Table 17-3 Declined Order Count measure details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: OrderHandlingFuture2	Name: Order_Cancelled
Associate event to trigger	Event Location: CancelOrderandSendNotification_InputCriterion	Select: CancelOrderandSendNotification_InputCriterion ENTRY
Create counter	Counter Location: OrderHandlingFuture2	Name: Declined Order Count
Associate trigger to counter	Counter Controls: OrderHandlingFuture2	Trigger: Order_Cancelled Resulting action: Add One
Add measure	Data Mart Model → Measures: OrderHandlingFuture2 Cube	Name: Declined Order Count Source: Declined Order Count Fact Aggregation Function: Sum

Order Price Total

This measure aggregates the price of all orders submitted (Table 17-4).

Table 17-4 Order Price Total measure details

Activity	Description / Location	Name / Value
Create trigger	Trigger Location: OrderHandlingFuture2	Name: OrderVariable_Changed
Associate event to trigger	Trigger Source: OrderHandlingFuture2	Select: OrderVariable_CHANGED

Activity	Description / Location	Name / Value
Create metric	Metric Location: OrderHandlingFuture2	Name: Order Price Type: Decimal Default Value: 0.00
Associate trigger to metric	Trigger Location: OrderHandlingFuture2	Metric Value Maps: Trigger: OrderVariable_Changed Expression: OrderVariable_CHANGED/extendedData /variableData/TotalPrice
Add measure	Data Mart Model → Measures: OrderHandlingFuture2 Cube	Name: Order Price Total Source: Order Price Fact Aggregation Function: Sum

Order Price Average

This measure calculates the average order price for all orders submitted (Table 17-5).

Table 17-5 Order Price Average measure details

Activity	Description / Location	Name / Value
Add measure	Data Mart Model → Measures: OrderHandlingFuture2 Cube	Name: Order Price Average Source: Order Price Fact Aggregation Function: Average

Creating dimensions

In this section we create the dimensions for location and classification. Classification is the new dimensions for Future 2 to categorize the customer as REGULAR, SILVER, or GOLD.

Location

To create the dimensions for location, refer to “Creating a dimension” on page 349.

Classification

We create the customer classification from the OrderVariable as shown in Table 17-6.

Table 17-6 Classification metric details

Activity	Description / Location	Name / Value
Create metric	Metric Location: OrderHandlingFuture2	Name: Order Classification Type: String Length: 20 Default Value: 'REGULAR'
Associate trigger to metric	Trigger Location: OrderHandlingFuture2	Metric Value Maps: Trigger: OrderVariable_Changed Expression: OrderVariable_CHANGED/extendedData/variableData/Order/Classification
Add dimension	Data Mart Model → Dimensions: OrderHandlingFuture2 Cube	Name: Classification Attribute: classification Source: Order Classification

The Dimensions view after creating all the measures is shown in Figure 17-1.

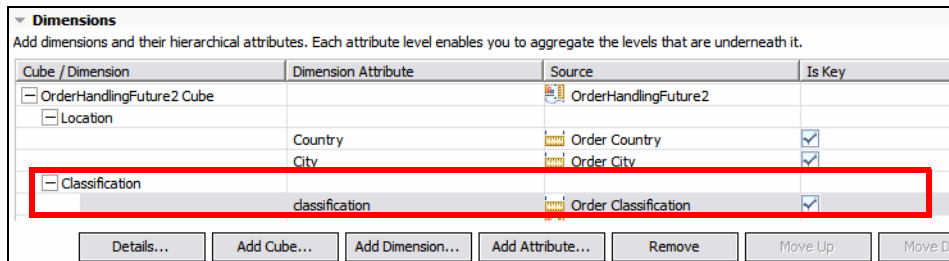


Figure 17-1 Dimensions for Future 2

Creating KPIs

To create the two KPIs, follow the instructions in “Creating KPIs” on page 353.

- ▶ Average Shipped Order Fulfillment Time
- ▶ Shipped Order %

Creating alerts

To create the two event definitions and alerts, follow the instructions in “Creating alerts” on page 362.

- ▶ Elapse Duration Alert
- ▶ ShippedOrdersLessThan85

Measuring duration times and decision percentages

For future 2 process we do not be implementing the round trip to the Modeler, but we create measures for the human task durations and one branch percentage.

Determine activity duration

We create measures to analyze the activity duration measure for Review Order and Ship Order to Customer. Follow the instructions in “Business measures for human task processing time” on page 377.

Determine branch percentages

We implement the branch percentage measure for Check Order Handling Policy for Automatic Approval (Yes) by counting the number of orders that go though the Check Customer Credit out of all the orders submitted.

Follow the instructions in “Determining branch percentages” on page 371, however, use the trigger for the Check Customer Credit entry. We named the metric and measure different: Check Order Handling Policy for Automatic Approval %.

Monitoring model content

Let us review the triggers, metrics, and measures that we defined (Table 17-7).

Table 17-7 Recapitulation of model elements

Component	Triggers	Metrics	Measures
OrderHandling Future1	<ul style="list-style-type: none"> ▶ CheckCustomer Credit_Entry ▶ <i>Order_Cancelled</i> ▶ Order_Entry ▶ Order_Exit ▶ Order_Shipped ▶ OrderVariable_ Changed 	<ul style="list-style-type: none"> ▶ CheckOrder Handling Policy for Automatic Approval % ▶ Order City ▶ <i>Order Classification</i> ▶ Order Country ▶ Order Number ▶ Order Price ▶ Shipped Flag ▶ Shipped Order % ▶ Start Time 	<ul style="list-style-type: none"> ▶ Order Count ▶ <i>Shipped Order Count</i> ▶ <i>Declined Order Count</i> ▶ Order Price Total ▶ Order Price Average ▶ Order Fulfillment Time ▶ Shipped Order % ▶ CheckOrder Handling Policy for Automatic Approval % <p>Dimensions:</p> <ul style="list-style-type: none"> ▶ Location ▶ <i>Classification</i> ▶ Shipped Flag ▶ Start Time

Component	Triggers	Metrics	Measures
UpdateOrder Database...	<ul style="list-style-type: none"> ▶ Elapsed_Duration_Time ▶ Shipped_Time 	<ul style="list-style-type: none"> ▶ Fulfillment Time ▶ Order ShippedTime ▶ Elapsed Duration Alert 	
ReviewOrder_ xxxxxxxx.	<ul style="list-style-type: none"> ▶ ReviewOrder_Entry ▶ ReviewOrder_Exit 	<ul style="list-style-type: none"> ▶ ReviewOrderProcessTime 	<ul style="list-style-type: none"> ▶ Review Order Process Time
ShipOrdertoCu stomer_ xxxxx	<ul style="list-style-type: none"> ▶ ShipOrdertoCusto mer_Entry ▶ ShipOrdertoCusto mer_Exit 	<ul style="list-style-type: none"> ▶ ShipOrdertoCustomerProcess Time 	<ul style="list-style-type: none"> ▶ Ship Order to Customer Process Time

In addition, we have defined two KPIs: Average Shipped Order Fulfillment Time (3 day target) and Shipped Order % (90% target).

The main difference from the implementation in Future 1 (Table 12-19 on page 383) is that we defined more triggers, metrics, and measures at the process level. We only have a few elements in the Update Order Database task. The new measures at the process level are in *italic*.

Deploying the model to the Monitor test environment

Deploy the monitor model to the monitor unit test environment by following the same steps as for Future 1 (refer to “Testing the monitor model” on page 392).

Important: You must undeploy the Future 1 model from the Monitor test environment before deploying the Future 2 model. This is because some activities have the same name. Follow the instructions in “Undeploying the monitor model in the test environment” on page 396 to remove the Future 1 model.

Testing the Future 2 monitor model

Note that by default the Monitor test environment runs on a different port. Therefore the Web service call has to be configured for the correct port. If the port is not changed, the Web service call fails for orders that are automatically approved. Orders that go through the order manager work fine.

Changing the port for the Web service call

Refer to “Changing the port number” on page 587 for instructions on how to configure the Monitor Server test environment to use port 9080.

To change the port in the application, open the Assembly diagram, select the checkCredit import and in the Properties view, Binding tab, change the port in the Address field.

In the development environment, we can deploy the model to the Monitor test server and test that the events are processed.

After deploying the applications and the Monitor model to the test environment, you can submit a few orders and process them. Then start the dashboard (select the server and *WBM Web Dashboard*) and select the model, monitoring context, and business measures (Figure 17-2).

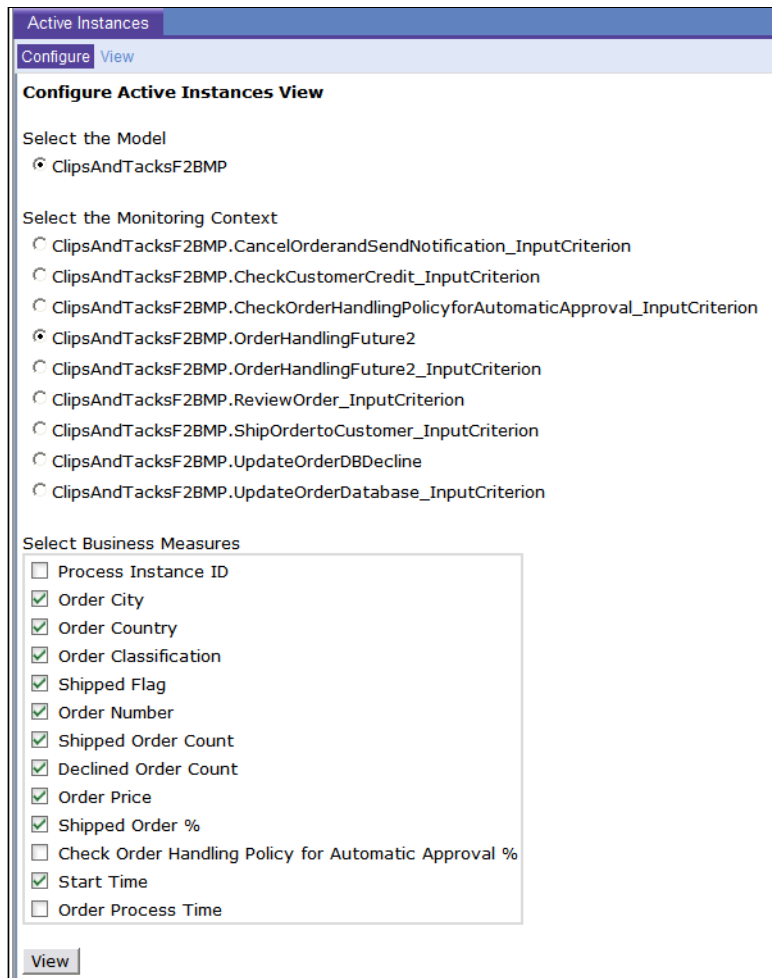
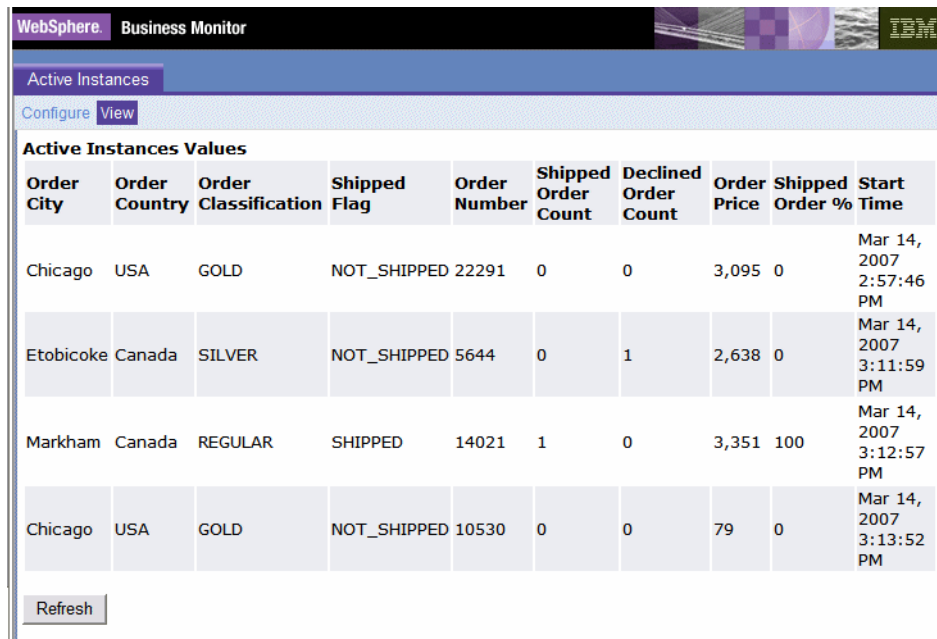


Figure 17-2 Monitor Dashboard Future 2 test environment

The active instances are displayed (Figure 17-3).



The screenshot shows the 'Active Instances Values' table in the WebSphere Business Monitor interface. The table has columns for Order City, Order Country, Order Classification, Shipped Flag, Order Number, Shipped Order Count, Declined Order Count, Order Price, Shipped Order %, and Start Time. There are four rows of data representing different instances.

Order City	Order Country	Order Classification	Shipped Flag	Order Number	Shipped Order Count	Declined Order Count	Order Price	Shipped Order %	Start Time
Chicago	USA	GOLD	NOT_SHIPPED	22291	0	0	3,095	0	Mar 14, 2007 2:57:46 PM
Etobicoke	Canada	SILVER	NOT_SHIPPED	5644	0	1	2,638	0	Mar 14, 2007 3:11:59 PM
Markham	Canada	REGULAR	SHIPPED	14021	1	0	3,351	100	Mar 14, 2007 3:12:57 PM
Chicago	USA	GOLD	NOT_SHIPPED	10530	0	0	79	0	Mar 14, 2007 3:13:52 PM

Figure 17-3 Monitor Dashboard Future 2 test environment instances

Once the monitor model is tested and validated, export the monitor model EAR file to be deployed to the WebSphere Monitor Server.

Deploying the Future 2 monitor model

We set up the Monitor Server in the same way as described in “Deploying the monitor model” on page 416.

Install the exported Monitor EAR file and run the seven steps of the setup wizard to deploy the monitor model:

1. Run Data Service Generation
2. Run schema Create Scripts
3. Run DMS Create Scripts
4. Import DB2 Cube Definition
5. Manage Alphablox cube
6. Configure CEI Distribution
Configure and run the scripts on Process Server to enable CEI events
7. Configure CEI Server Reboot

Measuring the Future 2 application

We used a script similar to the script described in “Measuring the Future 1 application” on page 477 to submit a number of orders to the Future 2 application.

We also tracked the process instances in the Monitor databases. Notice that new tables are generated for the Order Handling (Future 2) process.

Finally, we used the Monitor Dashboard to configure and view the KPIs and measures, and we performed dimensional analysis of the process data.

Monitor Dashboard for the Future 2 application

In this section we look at the different views that we configured for our revised application. Our dashboard is basically the same as for Future 1, with additional pages for the classification dimension and reports.

Gauge view

Let us start with the Gauge view for the key performance indicators (Figure 17-4). A first glance shows us that we reached our targets:

- ▶ The average order fulfillment time is about 2 days and 20 hours.
- ▶ Over 90% of orders have been shipped

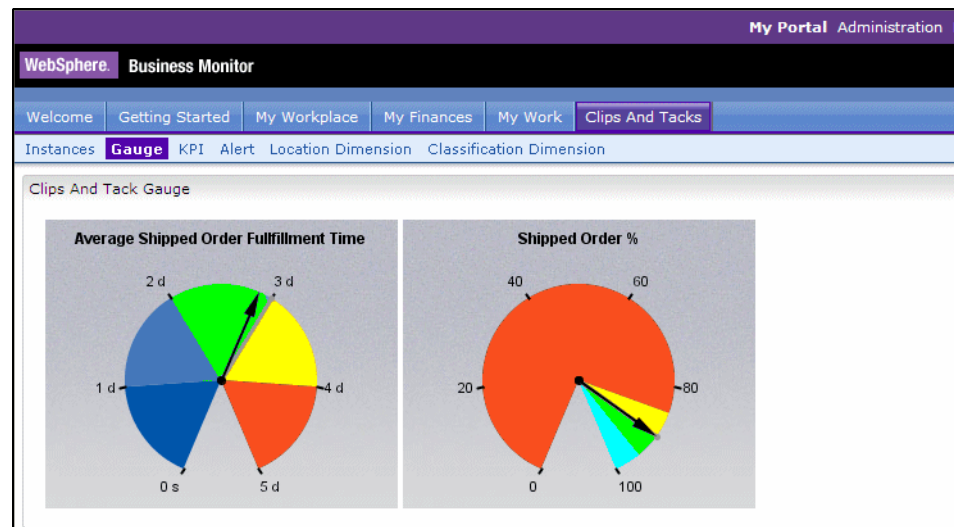


Figure 17-4 KPI gauges of revised process

KPI view

The KPI view shows us the exact figures (Figure 17-5). We shipped **90.2%** of the orders in an average time of **2 days 21 hours**.

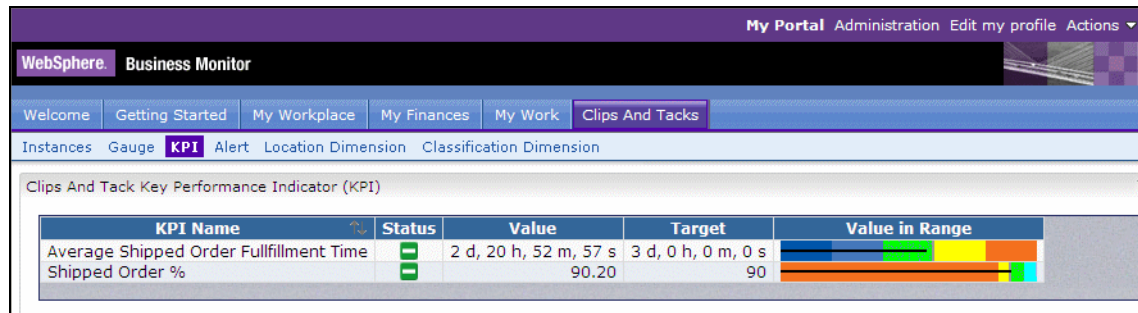


Figure 17-5 KPIs

Instances view

The Instances view shows 10 instances at a time, sorted by order price (Figure 17-6).

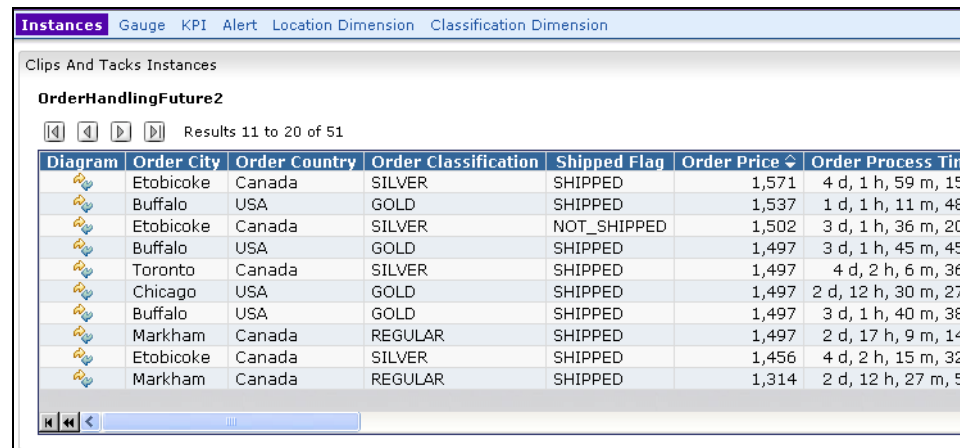


Figure 17-6 Instances

Dimensions view

We configured two dimensions: Location and classification.

Location dimension

First we can look at all orders (Figure 17-7). Notice the two scales for different measures.

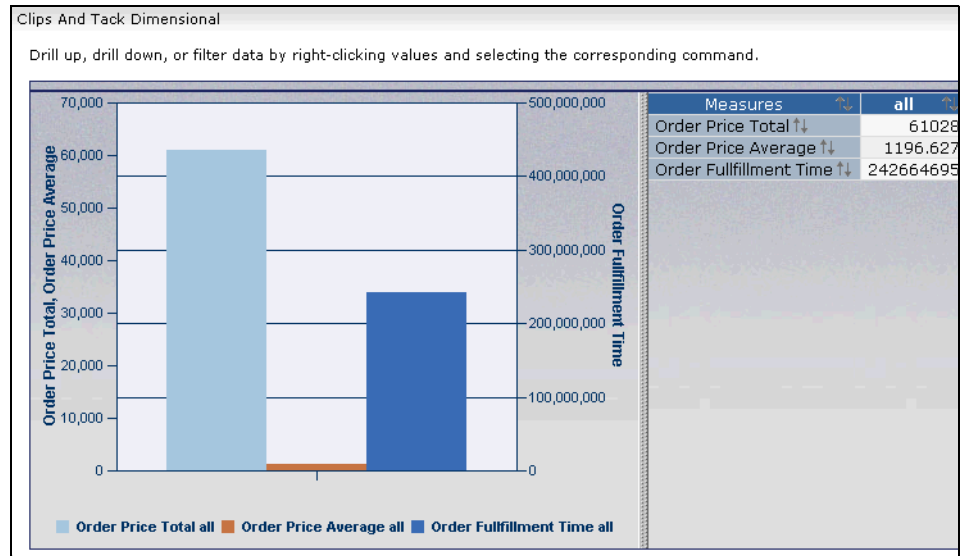


Figure 17-7 Orders by location: All

Select *all* (top right) and *Drill Down*. Now we see the data for the two countries (Figure 17-8).

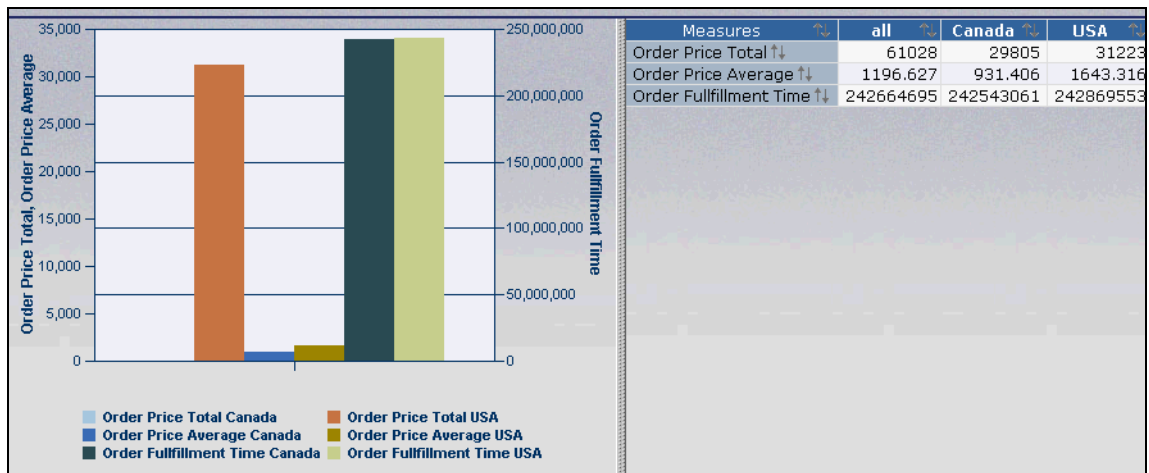


Figure 17-8 Orders by location: Country

Select *USA* and *Drill Down*, and the data changes to the cities within the USA (Figure 17-9).

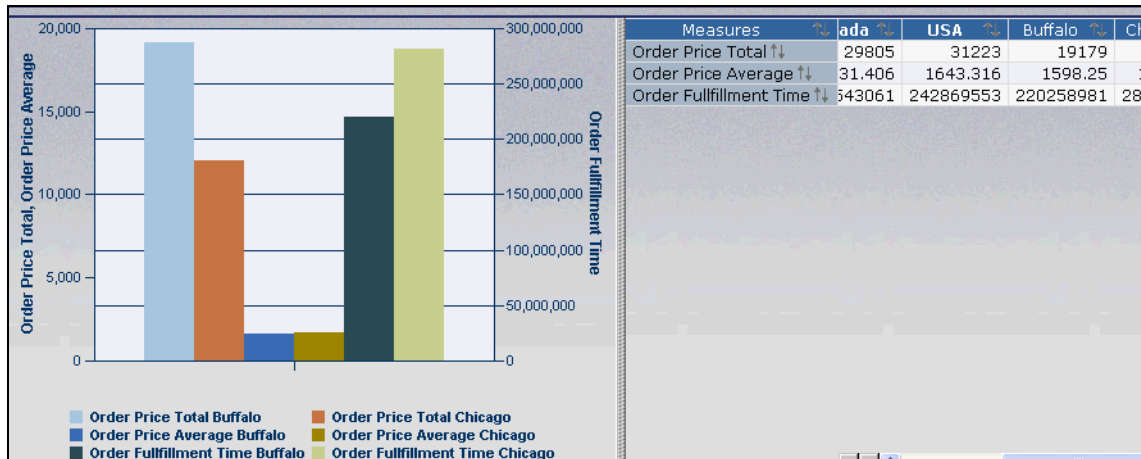


Figure 17-9 Orders by location: City

You can reconfigure the dimensional view in many ways. For example, you can select other measures to be displayed, and the type of chart used for the graph:

- ▶ Select one of the measures (Order Price Total) and *Member Filter*.
- ▶ Select the measures to be displayed: Order Count, Shipped Order Count, and Declined Order Count.
- ▶ Reconfigure the graph: Right-click in the graph area and select *Chart Types*.
- ▶ In the dialog select *Vertical Bar, Side-By-Side, 3D Effect* for Chart Type.
- ▶ Select the *Axis Placement* tab.
- ▶ For Measures, select *x-axis*.
- ▶ For Order Count and other measures, select *1st y-axis*.
- ▶ Click *OK*.

Figure 17-10 shows the Order Count, Shipped Order Count, and Declined Order Count by country.

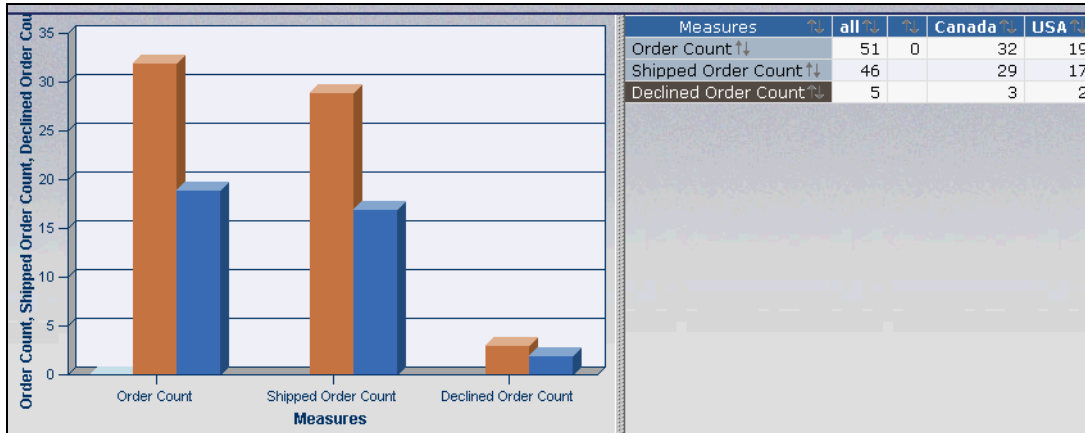


Figure 17-10 Order counts by location: 3d bar chart

Classification dimension

In the revised application we added the customer classification (REGULAR, SILVER, GOLD) to the process model and defined this classification as a dimension. Therefore, we can now display the business measures as a function of the classification.

The Monitor Dashboards enables us to configure views that show multiple measures based on the classification.

Figure 17-11 shows the Shipped Order % and Order Price Total by classification.

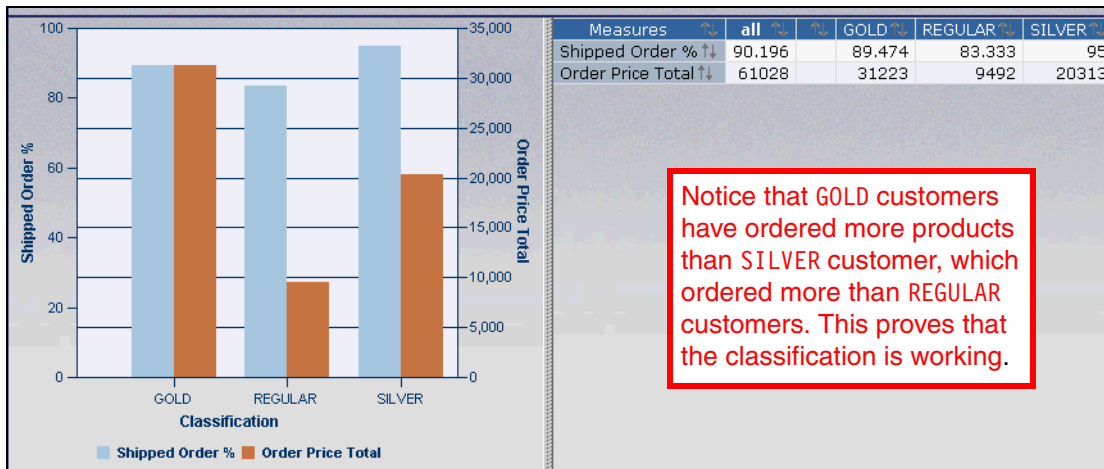


Figure 17-11 Shipped Order % and Order Price Total by classification.

Figure 17-12 shows the automatic approval rate by classification, indicating that orders from REGULAR customers are not automatically approved as often.

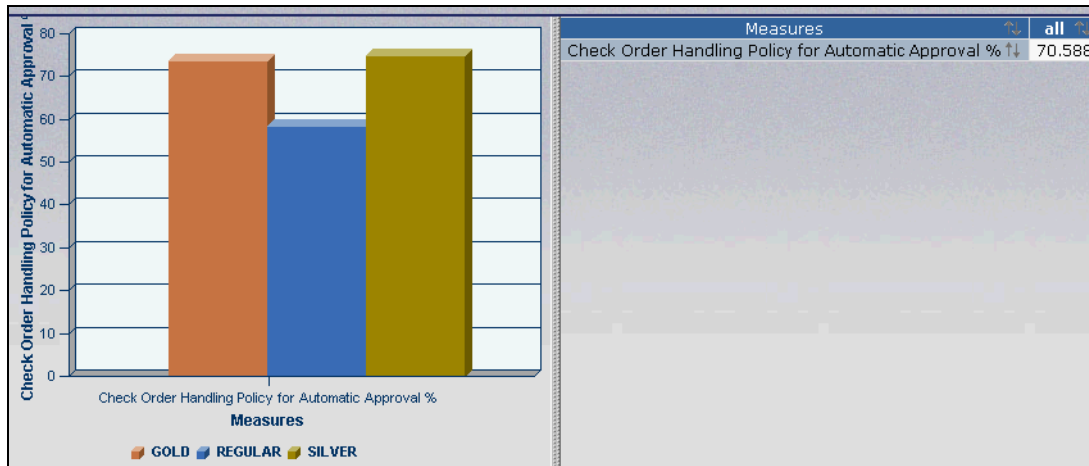


Figure 17-12 Automatic approval rate by classification

The Dimensions view can be configured with different graphics and also a number of different styles for each graphic.

Alerts view

We configured the Adaptive Action Manager as described in “Configuring the Adaptive Action Manager” on page 441. The Alerts view (Figure 17-13) shows an alert because of a slow shipment of an order.

Clips And Tack Alerts			
<input type="button" value="Mark Read"/> <input type="button" value="Mark Unread"/> <input type="button" value="Remove"/>			
<input type="button" value="Home"/> <input type="button" value="Previous"/> <input type="button" value="Next"/> <input type="button" value="End"/> Results 1 - 2 of 2			
<input type="checkbox"/>	Date and Time	Subject	Alert Source
<input type="checkbox"/>	Apr 7, 2007 4:13:32 PM	Order exceeded target shipment time	BA
<input type="checkbox"/>	Apr 8, 2007 7:32:29 AM	Too few orders shipped	

Figure 17-13 Alerts view

Click on the subject and the detailed message is displayed showing the subject and the body with the variable data added (Figure 17-14).

Clips And Tack Alerts

Details:

Date and time:
Apr 7, 2007 4:13:32 PM

Subject:
Order exceeded target shipment time

Business situation name:
Shipment is delayed

Body:
Order 25048 exceeded target shipment time.

Clips And Tack Alerts

Details:

Date and time:
Apr 8, 2007 7:32:29 AM

Subject:
Too few orders shipped

Business situation name:
Shipment Orders percentage too low

Body:
The shipment percentage is 84.62 (below 85)

Figure 17-14 Alert details with variable data

Reports view

The reports view enables us to see trends in data over time. Figure 17-15 shows a simple example with total orders and shipped orders, by classification, over several days (our limited data only shows two days).

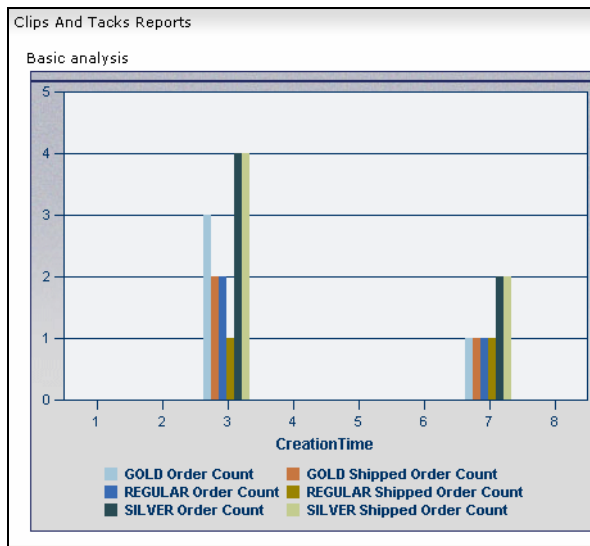


Figure 17-15 Reports view by classification

Figure 17-16 shows the order fulfillment time, by location, over several days.

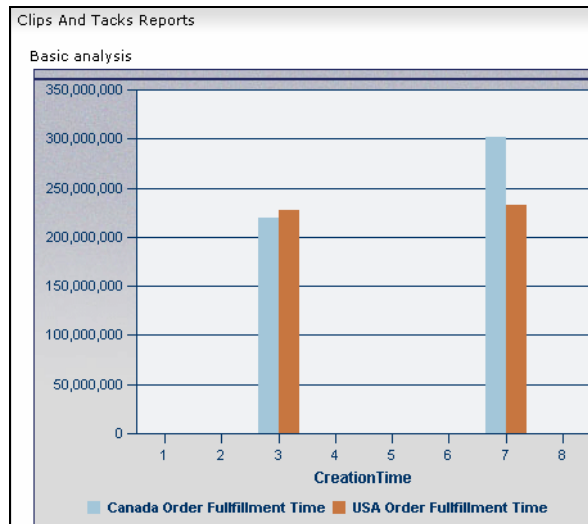



Figure 17-16 Reports view by location

Wiring two portlets on one portal page

Some portlets can be wired together so that an action in one portlet changes the display in the other portlet.

One example of such wiring is a portal page with the Alert and the Instances portlets wired together. The steps to create such a page with two portlets are outlined here:

- ▶ Create a portal page (for example, Alert Instance).
- ▶ Add the copied Clips And Tacks Alert and Instances portlets to the page (for example in vertical arrangement).
- ▶ When editing the page layout, select the *Wires* tab (Figure 17-17):
 - Select the source portlet (*Alert*) and sending the instance.
 - Select the target portlet (*Instances*) and receiving an instance.
 - Select *Private* (one user) or *Public* (all users) for the wire type.
 - Click  to add the wire.
 - Click *Done*.
- ▶ Configure both portlets (what data to display, colors).

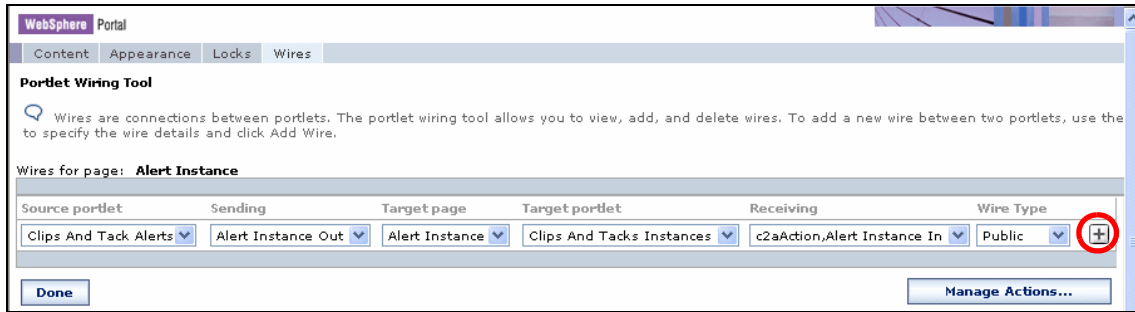



Figure 17-17 Wiring two portlets together

When displaying this Dashboard page you can click  in the Alert Source column to display the instance that caused the alert (Figure 17-18). Click *Reset* and by default all instances are displayed in the Instances portlet.

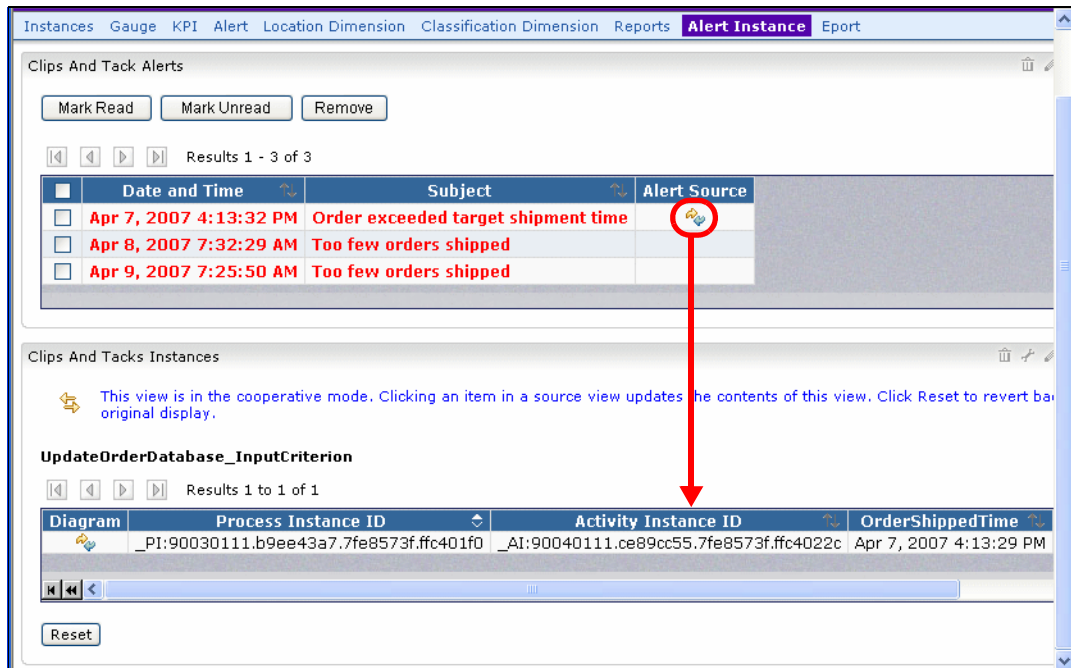


Figure 17-18 Wired portlets in action

Notice that the other alert type does not give you the option to show an instance because the alert is based on a KPI value and not data of one instance.

Using the visual model

The monitor model exported from the Modeler contains two SVG diagram files and two visual models in the .mm file. The two visual models are identical but will be used to visualize different measures:

- ▶ Process model to visualize instance data
- ▶ KPI model to visualize KPI values

Both visual models contain a section `<actions/>` where we can add actions to visualize instance metrics and KPIs. Two of the possible actions are:

- ▶ **setText**—Display the value of an instance metric or KPI

```
<setText textValue="value" textColor="color"
  <shapeSet ref="reference to shape label" />
</setText>
```

- ▶ **setColor**—Change the color of a shape or connection

```
<setColor condition="condition" outlineColor="color">
  <shapeSet ref="reference to shape or connection" />
</setColor>
```

Displaying instances in the visual model

To display instance metrics and the path that a process instance took through the model, we use `<setText>` and `<setColor>`.

Displaying instance metrics

For example, we want to display the order price and the customer classification on the top and bottom of the Check Order handling for Automatic Approval task. Here is the coding:

```
<actions>
  <setText textValue="concat('Price: ', Order_Price)" textColor="green">
    <shapeSet ref="Check_Order_Handling_Policy_for_Automatic_Approval
      _top_label" />
  </setText>
  <setText textValue="concat('Classification: ', Order_Classification)"
    textColor="green">
    <shapeSet ref="Check_Order_Handling_Policy_for_Automatic_Approval
      _bottom_label" />
  </setText>
</actions>
```

In this code `Order_Price` and `Order_Classification` are the instance metrics defined in the monitor model.

The reference to the shapes refer to existing IDs of <shapeSet> entries in the visual model:

```
<shapeSet displayName="top label for Check_Order_Handling_xxx"
  id="Check_Order_Handling_Policy_for_Automatic_Approval_top_label">
  <shapes>18 </shapes>
</shapeSet>
```

In the same way we can display other metrics (customer country, city, and credit rating) as the labels of other tasks.

Color coding the path of a process instance

For a process instance, we want to color code the path in blue for completed tasks and in red for active tasks (human tasks).

We have to define new metrics that can be tested in the <action> code. For each task we define a status metric with a value of none (not yet started), started (a human task has started), and completed (a task is complete). We defined these metrics:

- ▶ StatusAutoApproval—none or completed
- ▶ StatusCreditCheck (subprocess)—none, started, or completed
- ▶ StatusReview and StatusShip (human tasks)—none, started, or completed
- ▶ StatusDBShip (database update) and StatusCancel—none or completed

Marking a completed task in blue

To mark a completed task and the in and out connections in blue, we use this code:

```
<setColor condition="StatusAutoApproval != 'none'" outlineColor="blue">
  <shapeSet ref="Connection" />
  <shapeSet ref="Check_Order_Handling_Policy_for_Automatic_Approval" />
  <shapeSet ref="Connection_x003a_2" />
  <shapeSet ref="Approve_Without_Review_x003f_" />
</setColor>
```

The connection IDs and decision blocks IDs come straight from the Modeler:

```
Connection:2 ==> Connection_x003a_2
Approve Without Review? ==> Approve_Without_Review_x003f_
```

Marking an active human task in red

Here is the code to mark an active human task:

```
<setColor condition="StatusReview = 'started'" outlineColor="red">
  <shapeSet ref="Review_Order" />
</setColor>
```

To completely color code the process instance path takes many <setColor> statements.

Displaying KPIs in the visual model

The original two KPIs would not make much sense to be displayed in the visual model. We can define a few new KPIs for this purpose:

- ▶ Review Process Time—KPI based on the Review Order Process Time measure
- ▶ Ship Process Time—KPI based on the Ship Order to Customer Process Time measure
- ▶ Review Queue—KPI of the number of process instances waiting for review, based on a ReviewQueue measure that is based on a ReviewQueue counter, which is increased by one on entry, and decreased by one on exit of the Review Order task
- ▶ Ship Queue—KPI of the number of process instances waiting for shipping

To display these KPIs in the visual model we use this coding:

```
<setText textValue="concat('Queue: ', ReviewQueue)" textColor="green">
  <shapeSet ref="Review_Order_2_top_label" />
</setText>
<setText textValue="concat('Duration: ', Review_Process_Time)"
  textColor="green">
  <shapeSet ref="Review_Order_2_bottom_label" />
</setText>
<setText textValue="concat('Queue: ', ShipQueue)" textColor="green">
  <shapeSet ref="Ship_Order_to_Customer_2_top_label" />
</setText>
<setText textValue="concat('Duration: ', Ship_Process_Time)"
  textColor="green">
  <shapeSet ref="Ship_Order_to_Customer_2_bottom_label" />
</setText>
```

Dashboard portal page to display the visual model

To display the visual model with both the KPI values and the process instance, we create a portal page with two portlets:

- ▶ Portal page: Instance Diagram
- ▶ Portlets: Clips And Tacks Instances, Clips And Tacks Diagram
- ▶ Wire: The two portlets are wired together as described in “Wiring two portlets on one portal page” on page 562.

Visual KPI model

When the Instance Diagram portal page is displayed, the top of the page displays the instances (same as “Instances view” on page 556), and the bottom part display the visual KPI model with the KPI values (Figure 17-19).

Clips And Tacks Instances

OrderHandlingFuture2

Results 1 to 17 of 17

Diagram	Order City	Order Country	Order Classification	Shipped Flag	StatusReview	StatusShip	Order Number
	Chicago	USA	GOLD	SHIPPED	completed	completed	10,657
	Toronto	Canada	SILVER	NOT_SHIPPED	completed	none	36,381
	Toronto	Canada	SILVER	SHIPPED	completed	completed	40,380
	Buffalo	USA	GOLD	NOT_SHIPPED	started	none	2,955
	Etobicoke	Canada	SILVER	SHIPPED	completed	completed	8,868
	Buffalo	USA	GOLD	SHIPPED	completed	completed	5,592
	Buffalo	USA	GOLD	SHIPPED	completed	completed	5,497
	Etobicoke	Canada	SILVER	SHIPPED	none	completed	9,521
	Chicago	USA	GOLD	SHIPPED	none	completed	14,722
	Chicago	USA	GOLD	NOT_SHIPPED	none	started	11,934
	Etobicoke	Canada	SILVER	SHIPPED	none	completed	3,565

Diagram (truncated)

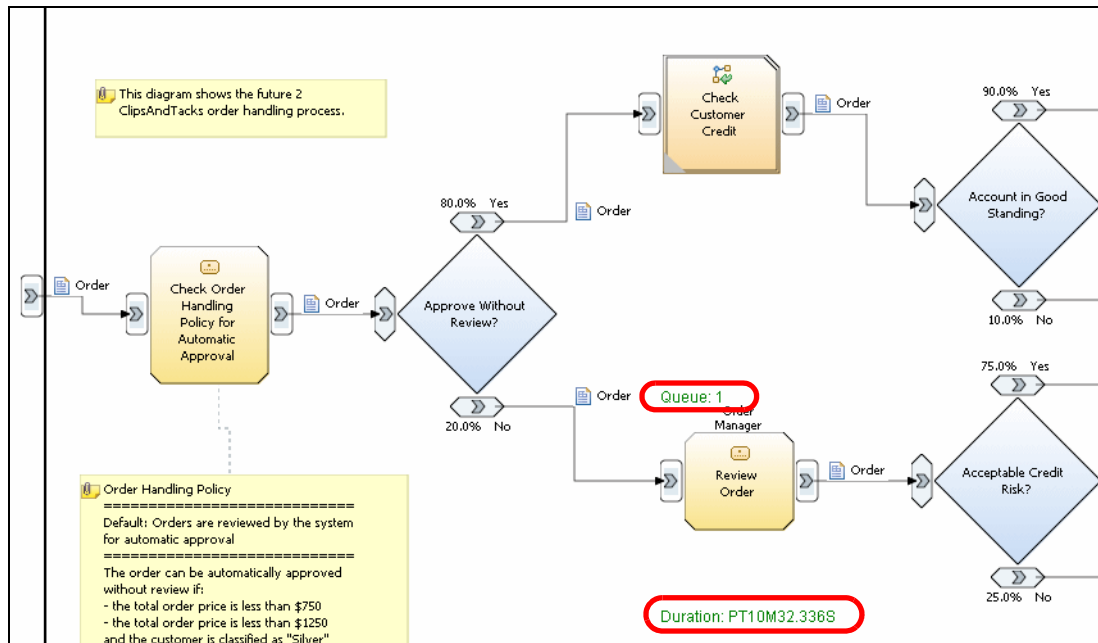


Figure 17-19 Instances and Visual KPI model

The right side of the diagram shows the queue and processing time of the Ship Order to Customer task (Figure 17-20).

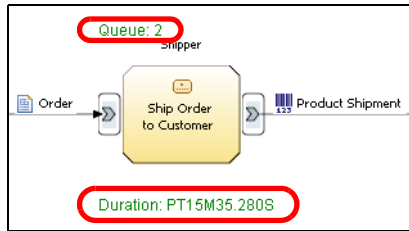



Figure 17-20 Ship Order to Customer queue and processing time

Visual process instance model

By clicking on the diagram link  in the instances list, one instance is displayed in the visual model with path and metrics (Figure 17-21).

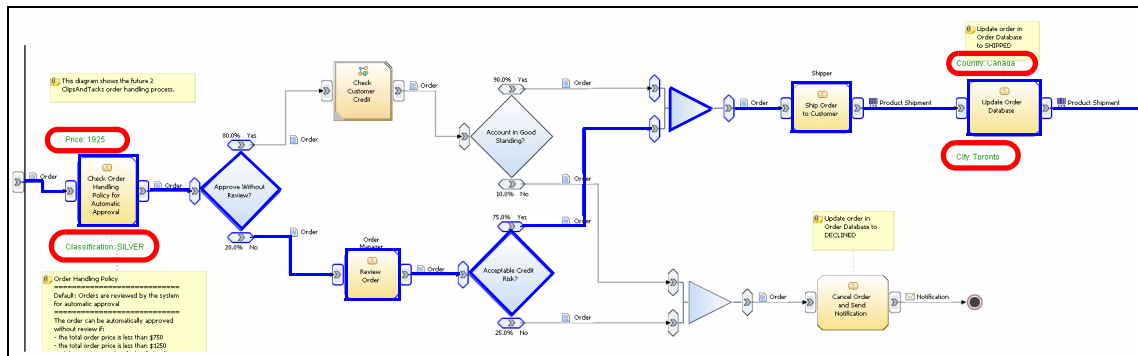


Figure 17-21 Process instance with blue path and metrics

An instance currently in review shows up as depicted in Figure 17-22.

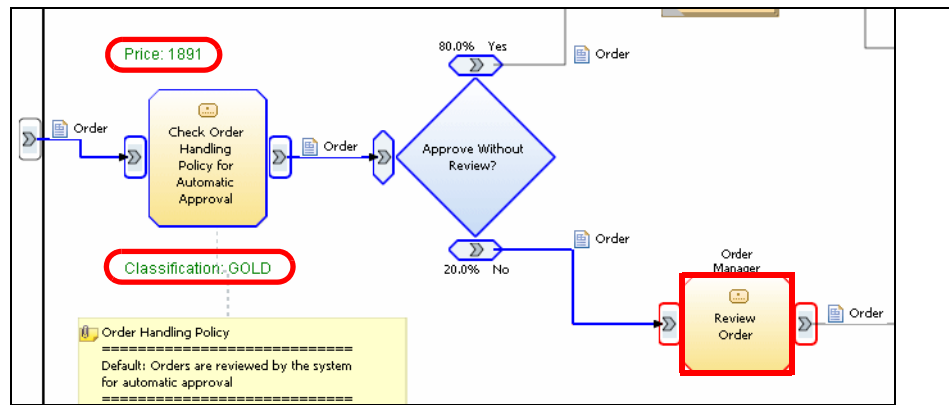


Figure 17-22 Process instance in review

An instance currently in shipping shows up as illustrated in Figure 17-22, with the new metrics for customer rating, country, and city.

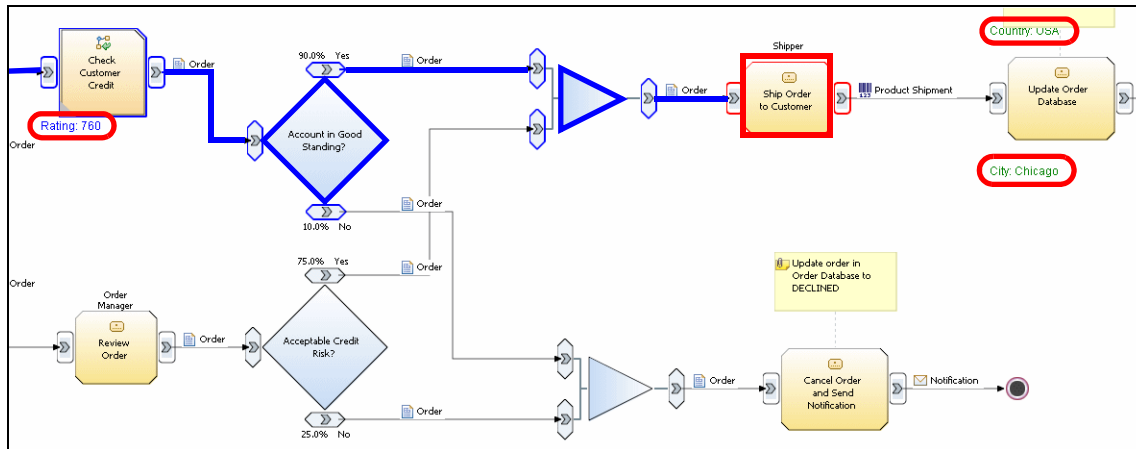


Figure 17-23 Process instance in shipping with metrics

We provide the monitor model with visual coding as a separate model in the ClipsAndTacksF2BMP project:

ClipsAndTacksF2BMPvisual.mm

Notice that there are no dialogs to define these actions. They must be coded in the .mm file directly. From this monitor model a deployable enterprise application can be generated, for example ClipsAndTacksF2BMPvisualEAR.

Installing a new version of the monitor model

If you only make small changes to a monitor model, you can install a new version of the model without changing or redeploying the monitor model application itself:

- ▶ Changes that are allowed include adding a metric, adding a process, and adding a dimension. Such changes can be deployed using a new version of the monitor model.
- ▶ Changes that require a complete new monitor model to be deployed include changing a dimension, changing the data type of a metric, and changing the process model or the activities.

Deploying a version of a monitor model must be carefully executed. We suggest that you take a backup copy of the configuration and the two monitor databases before starting the deployment.

Steps to deploy a monitor model version

The process of deploying a monitor model version is outlined here:

- ▶ Make no changes to the base application.
- ▶ Change the monitor model as described above (small changes).
- ▶ Change the timestamp of the model (Figure 17-24). The timestamp must be newer than the previous model.

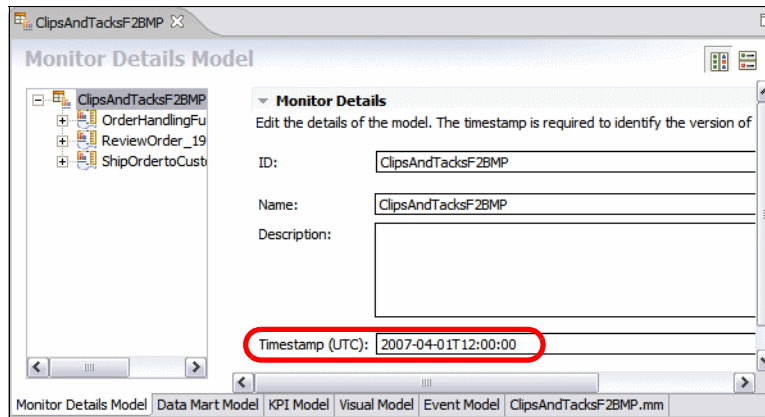


Figure 17-24 Monitor model timestamp

- ▶ Generate the monitor model EAR (select the model and *Generate Monitor EAR*). Select a different project name, such as, C1 ipsAndTacksF2BMP2EAR.
- ▶ Install the new EAR into the Monitor Server using the administrative console. This installation steps fails if the timestamp has not been changed.
- ▶ Select *Applications* → *Monitor Models*. Both versions of the model are shown (Figure 17-25).

Select	Model	Version	Startable	Application	Status
<input type="checkbox"/>	ClipsAndTacksF2BMP	2007-03-12T10:43:09	✓	ClipsAndTacksF2BMPEAR	➔
<input type="checkbox"/>	ClipsAndTacksF2BMP	2007-04-01T12:00:00	✗	ClipsAndTacksF2BMP2EAR	✗

Total 2

Figure 17-25 Monitor model versions

- ▶ Select the new version and run the setup wizard in the same way as for the first version.
- ▶ The generated DDL files update the current database schema, instead of producing a new schema.
- ▶ You have to proceed through all seven steps of deployment.
- ▶ Note that no new DB2 replicators are generated. The existing replicators work on the new version as well.
- ▶ The Dashboard works on the new model version. Some portlets may have to be reconfigured.

Summary

In this chapter we measured the revised application and used the Monitor Dashboard to verify that the application meets the expectations by exceeding both key performance indicators.

Using the customer classification as a dimension, we also analyzed the order data to get a better understanding if the classification idea is improving the order handling process.

ClipsAndTacks management is very happy about their investment into an improved application that meets customer requirements and streamlines their operation.



Part 6

Appendixes



A

Installation of the development products

In this appendix we describe how to install the base products of the WebSphere Business Integration suite.

Installation of WebSphere Business Modeler V6.0.2

To install the WebSphere Business Modeler V6.0.2, follow these steps:

- ▶ Start the setup.exe from the installation CD or folder.
- ▶ In the Launchpad, select *Install IBM WebSphere Business Modeler Advanced Version 6.0.2* (Figure A-1).



Figure A-1 WebSphere Business Modeler Launchpad

- ▶ In the Welcome page, click *Next*.
- ▶ Accept the license agreement and click *Next*.
- ▶ Enter the name of the installation directory. The default is:
C:\Program Files\IBM\Modeler6
C:\Modeler602 <==== our choice
- ▶ Click *Next* in the summary panel.
- ▶ Wait for the installation to complete.
- ▶ Optionally select *Run WebSphere Business Modeler* and click *Next*.
- ▶ Click *Finish* to exit the installation wizard.

Interim fixes and fix packs for the Modeler

Investigate if interim fixes or Fix Packs are available and install them.

Tip: You can find the latest list of recommended fixes for WebSphere Business Modeler Advanced at:

<http://www-1.ibm.com/support/docview.wss?rs=2025&uid=swg27008413>

Starting the Modeler

Start the Modeler using *Start → Programs → WebSphere Business Modeler → WebSphere Business Modeler*.

Select a location for the workspace, for example:

c:\Workspaces\Modeler602ClipsTacks

Installation of WebSphere Integration Developer V6.0.2

To install WebSphere Integration Developer V6.0.2 follow these steps:

- ▶ Start the `launchpad.exe` from the installation CD or folder.
- ▶ In the Launchpad, select *View the Installation Guide* if you require help for installing the product.
- ▶ In the Launchpad, select *Install IBM WebSphere Integration Developer V6.0.2* (Figure A-2).

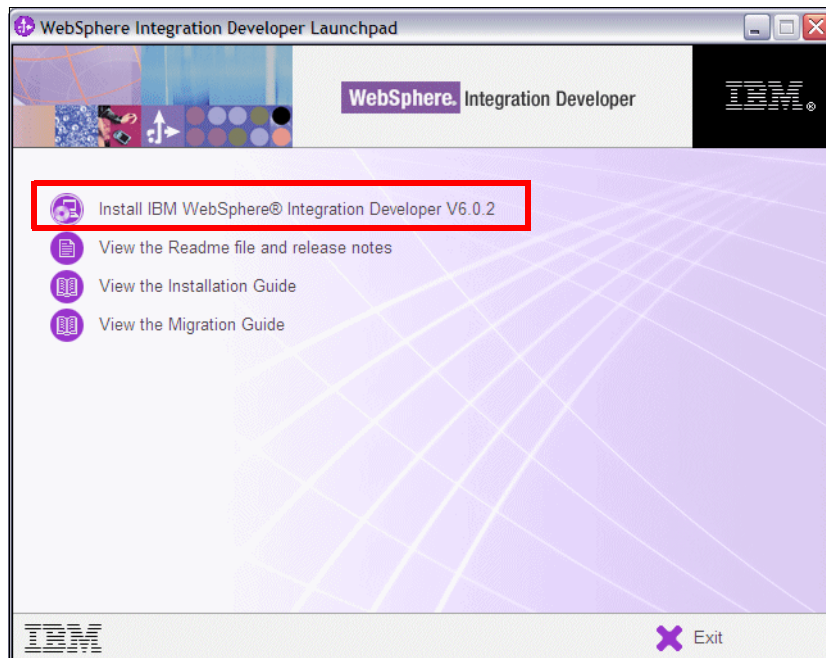


Figure A-2 WebSphere Integration Developer Launchpad

- ▶ In the Welcome panel, click *Next*.
- ▶ Accept the license agreement and click *Next*.
- ▶ Enter the name of the installation directory. The default is:
C:\Program Files\IBM\WebSphere\ID\6.0
C:\WID602 <==== our choice
- ▶ In the features panel, select *Integrated Test Environment* (Figure A-3).

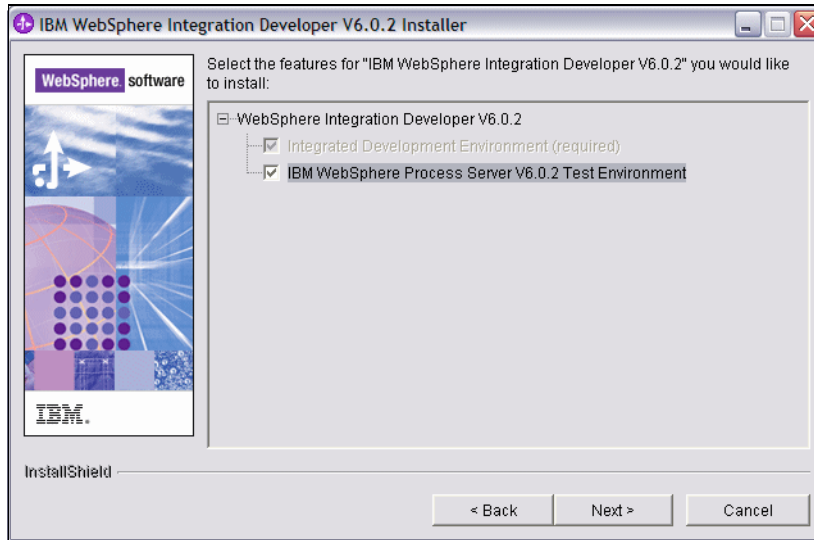


Figure A-3 Installing the integrated test environment

- ▶ Select the *WebSphere Process Server* profile (Figure A-4). For our scenario we do not require the WebSphere Enterprise Service Bus server.

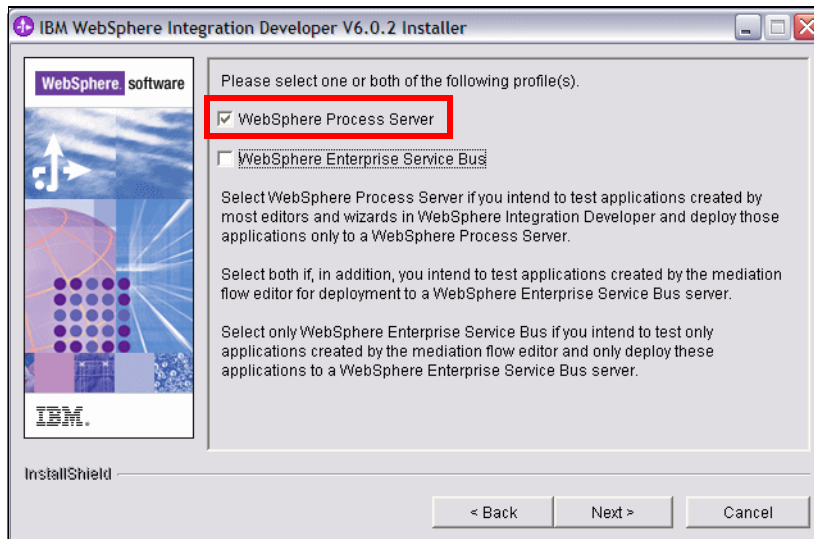


Figure A-4 Creating a profile for WebSphere Process Server

- ▶ Click *Next* in the summary panel.
- ▶ Be patient, the installation can take up to 2 hours.

- ▶ When the installation is finished, you receive a confirmation (Figure A-5). Click *Next*.

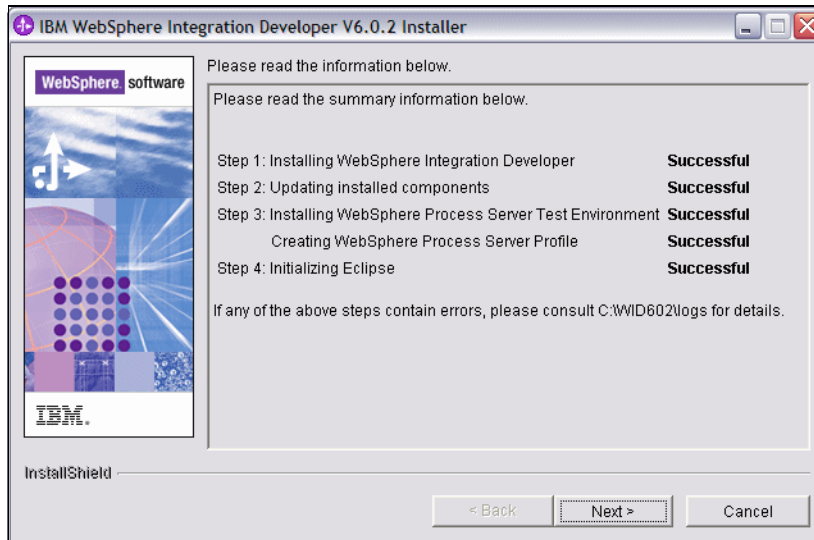


Figure A-5 Installation successful confirmation

Possible installation error with Data Execution Protection

License registration fails at the end of the installation when Data Execution Protection (DEP) is enabled in Microsoft Windows Enterprise Edition 2003 SP1 or Windows XP Professional SP2 with DEP compatible CPUs (Figure A-6).

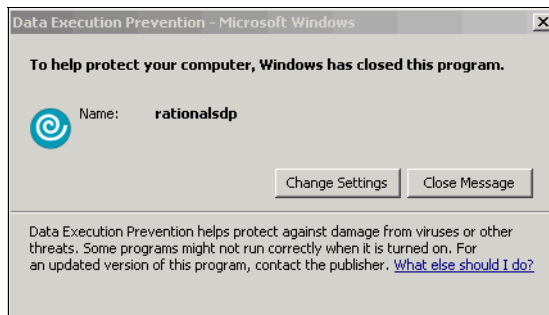


Figure A-6 Installation error with Data Exception Protection

If you receive this error, refer to this Web site for help:

http://www-1.ibm.com/support/docview.wss?rs=2043&context=SSCM5S&dc=DB520&uid=swg21225545&loc=en_US&cs=UTF-8&lang=en&rss=ct2043rational

Rational Product Updater

After a successful installation, we investigate if updates are available:

- ▶ Read the summary panel (successful installation) and click *Next*.
- ▶ Select *Launch Rational Product Updater* to verify if product updates are available (Figure A-7). Click *Finish*.

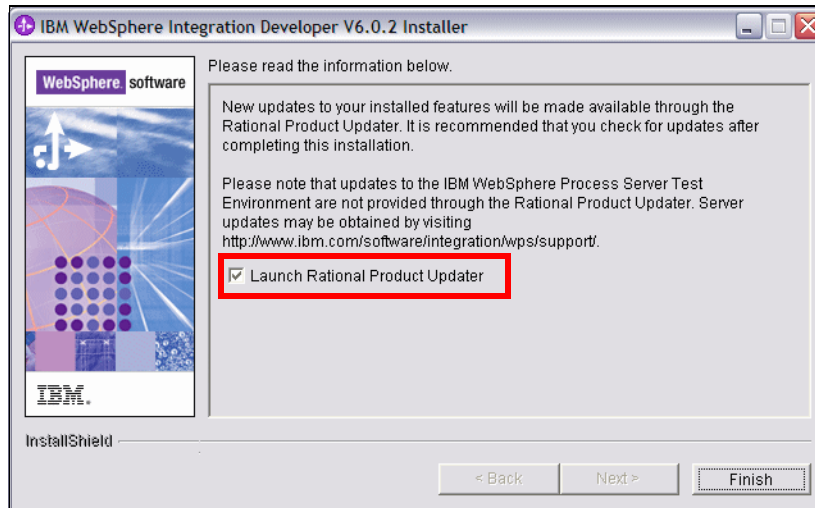


Figure A-7 Launch Rational Product Updater

- ▶ The Rational Product Updater opens. Select *IBM WebSphere Integration Developer* and click *Find Updates* (Figure A-8).

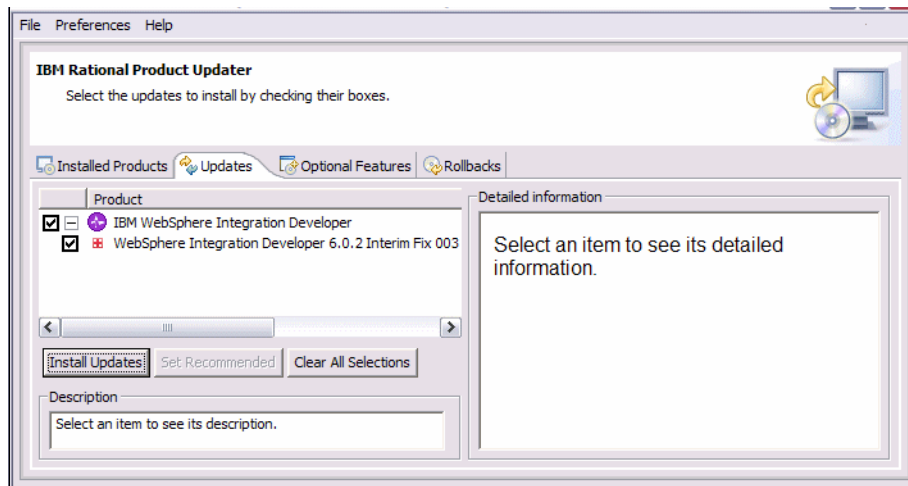


Figure A-8 Rational Product Updater

You can download the Interim fixes locally and install them through the Rational Product Updater.

You have to be connected to the Internet to check for product updates.

- ▶ If there are product updates available, install them. Then close the Rational Product Updater.
- ▶ Click *Exit* in the Launchpad.

Interim fixes and fix packs for Integration Developer

Investigate if interim fixes or fix packs are available and install them:

- ▶ WebSphere Integration Developer 6.0.2.2 is available as of August 1, 2007

Tip: You can find the latest list of recommended fixes for WebSphere Integration Developer at:

<http://www-1.ibm.com/support/docview.wss?rs=2308&uid=swg27006685>

Interim fixes and fix packs for the process server test environment

You may have to install fixes to the test environment. See “Interim fixes and fix packs for Process Server” on page 596 for further instructions.

Installing the information service feature

An IBM Information Service Activity Plug-In has been made available. It has to be installed for our scenario:

- ▶ Click *Optional Features* in Rational Product Updater. Select *IBM Information Service Activity* and click *Install Features* (Figure A-9).

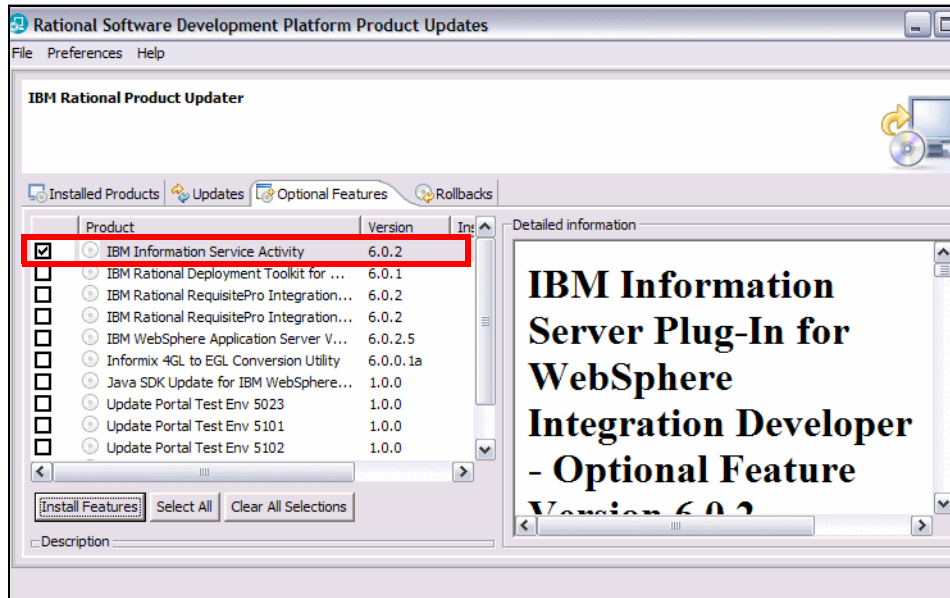


Figure A-9 Installing the information service feature

Starting WebSphere Integration Developer

You are now ready to start WebSphere Integration Developer using *Start* → *All Programs* → *IBM WebSphere* → *Integration Developer V6.0.2* → *WebSphere Integration Developer V6.0.2*.

See “Starting Integration Developer” on page 215 for further information.

Installation of the Monitor Development Toolkit

Prior to installation of the WebSphere Business Monitor V6.0.2 Development Toolkit, it is necessary to complete the following tasks:

- ▶ Install WebSphere Integration Developer V6.0.2 as described in “Installation of WebSphere Integration Developer V6.0.2” on page 578.
- ▶ Create a WebSphere Process Server user. This user must have administrative privileges.

To install the Monitor Development toolkit, follow these steps:

- ▶ Start the launchpad by running `setup.bat` from the installation CD or folder.
- ▶ In the Launchpad, click *Install* (Figure A-10)

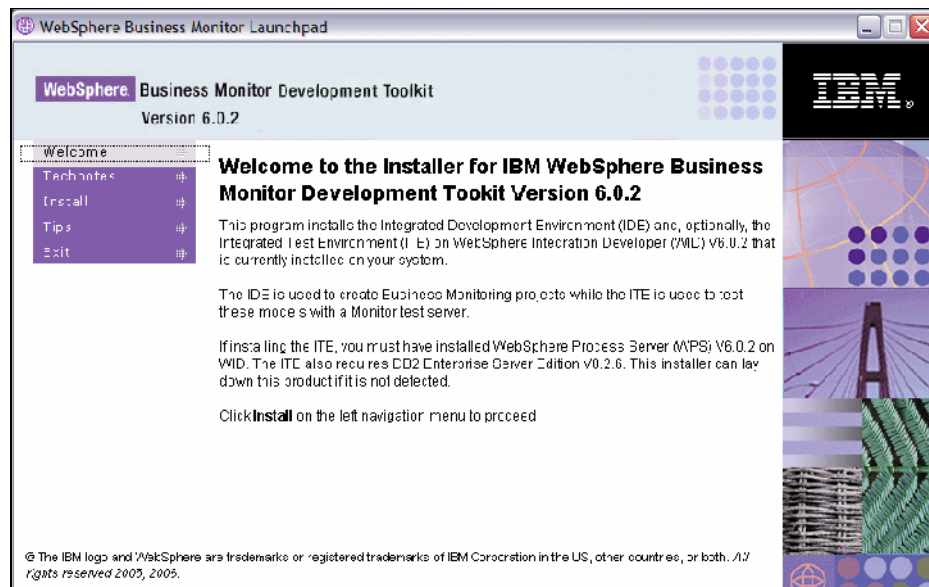


Figure A-10 WebSphere Business Monitor Development Toolkit: Launchpad

- ▶ Accept the license agreement.
- ▶ In the installation options window, select *Integrated Test Environment* and then click *Next* (Figure A-11).

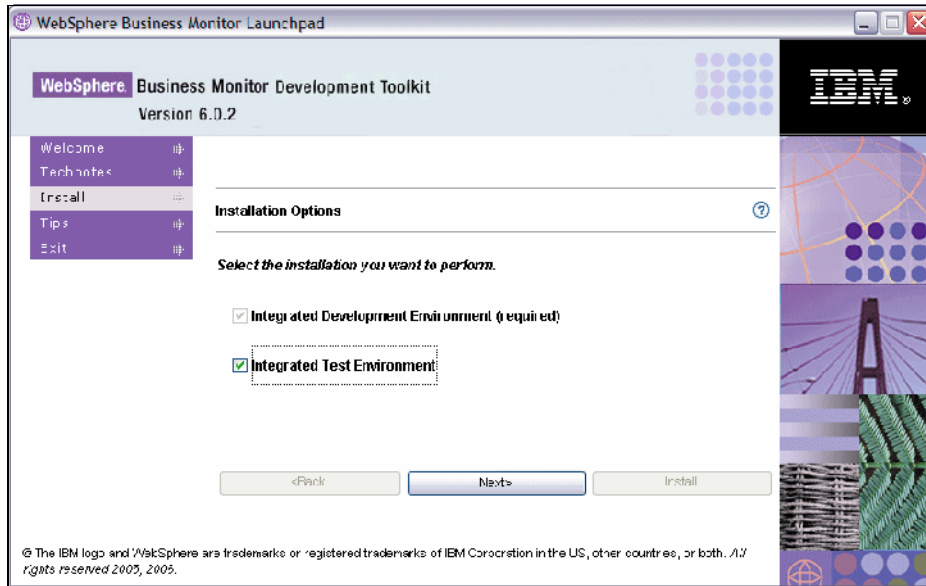


Figure A-11 Monitor Development Toolkit: Installation options

- ▶ In the Integrated Development Environment target directories panel, click *Next*.
- ▶ In the Integrated Test Environment target directories panel, enter the DB2 home directory (Figure A-12), then click *Next*:

C:\IBM\DB2\SQLLIB

<==== our choice

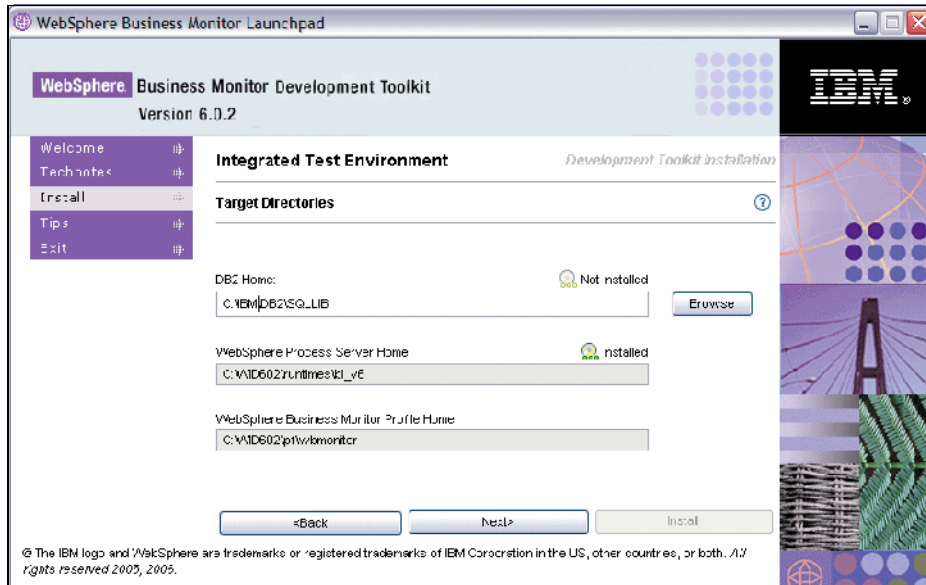


Figure A-12 Monitor Development Toolkit: Integrated test environment target directories

- ▶ Enter the user name and password for the DB2 and WebSphere Process Server users, then click *Next* (Figure A-13). The DB2 user will be created if it does not already exist.

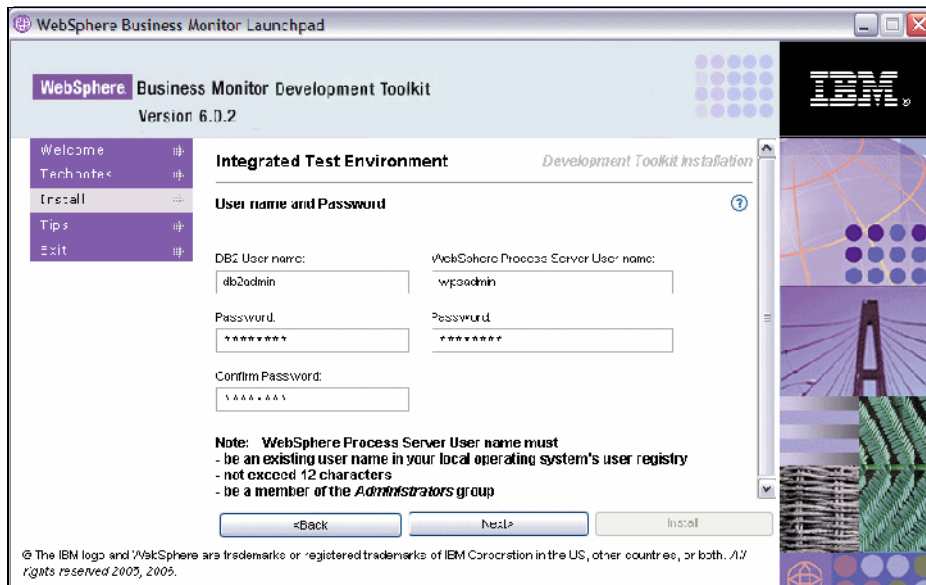


Figure A-13 Monitor Development Toolkit: Integrated Test Environment setup

- ▶ Review the Installation Summary and click *Install*. The installation takes approximately one hour.
- ▶ When the installation completes successfully, you will see confirmation of this (Figure A-14). Click *Finish* and then click *Yes* to close the Launchpad.

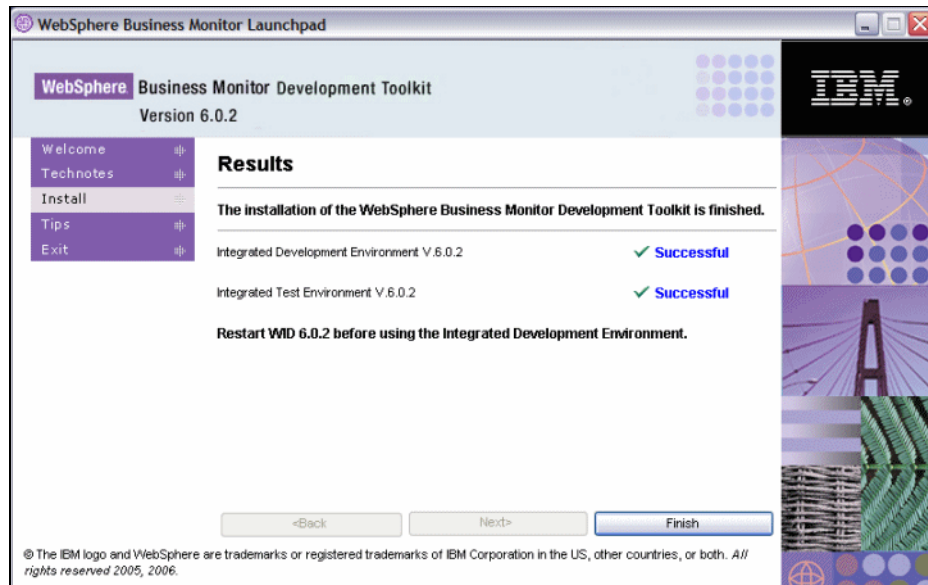


Figure A-14 Monitor Development Toolkit: Installation complete

Changing the port number

If you test business processes with Web services, then you may run into problems with the port number. By default, the Web services are configured with port 9080, but the Monitor test environment runs with 9081 (or similar).

If you do not plan to run the Process Server test environment and the Monitor Server test environment at the same time (which would need a 3-GB system) then you can change the port to 9080:

- ▶ Start the Monitor Server test environment.
- ▶ Open the administrative console
- ▶ Expand *Servers* → *Application Servers* and select *server1*.
- ▶ Scroll down to the Communications and select *Ports*.
- ▶ Click *WC_defaulthost* and change the port to 9080. Click *OK*.
- ▶ Expand *Environment* and select *Virtual Hosts*.
- ▶ Click *default_host*, then click *Host Aliases*.
- ▶ Click *New* and enter * and 9080. Click *OK*.
- ▶ Save the configuration and stop the server.

Installation of WebSphere Process Server V6.0.2

Note: We will enable security and configure the server to use LDAP. Therefore the user and password you use in this section must be valid for the LDAP server you installed in “Installation of IBM Tivoli Directory Server V5.2” on page 612.

In our example we use the user w i d.

To install the WebSphere Process Server V6.0.2, follow these steps:

- ▶ Start the launchpad.exe from the installation CD or folder.
- ▶ In the Launchpad, select *WebSphere Process Server installation* (Figure A-15).

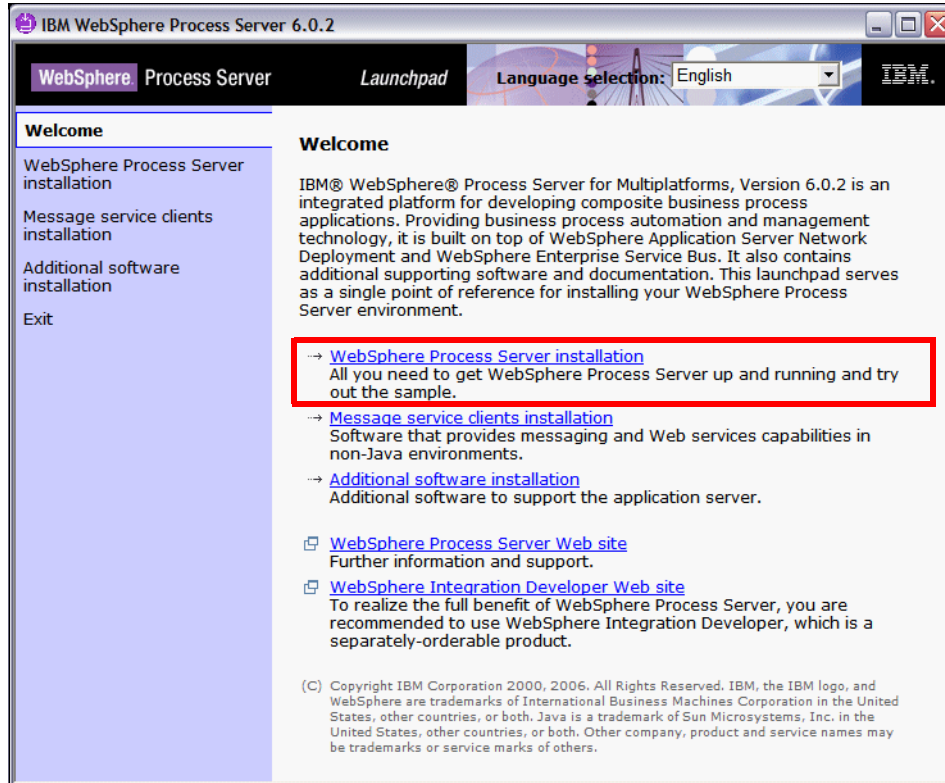


Figure A-15 WebSphere Process Server: Launchpad

- ▶ Select *Launch the Installation Wizard for WebSphere Process Server for Multiplatforms* (Figure A-16).

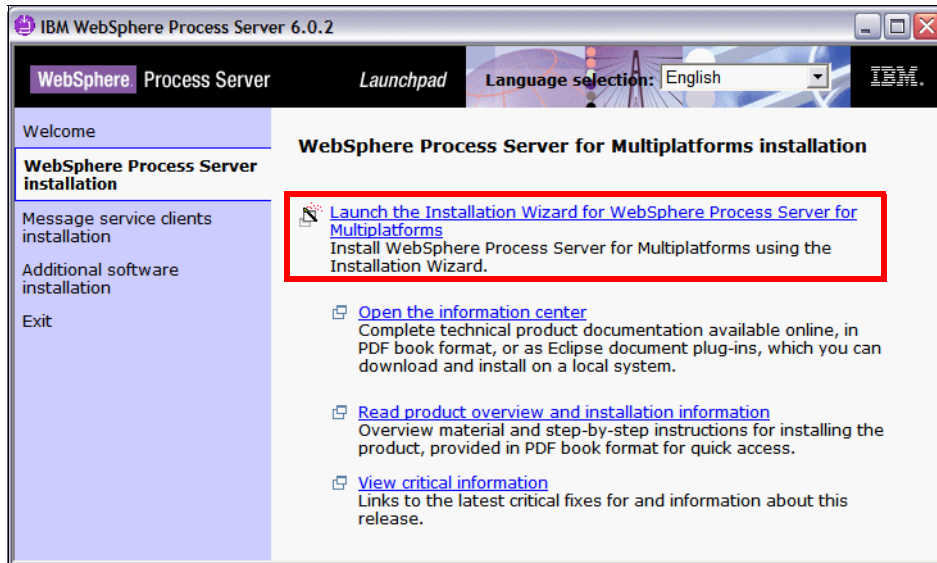


Figure A-16 WebSphere Process Server: Launch installation

- ▶ In the Welcome page, click *Next*.
- ▶ Accept the license agreement and click *Next*.
- ▶ In the System prerequisite check panel, click *Next*.

Note: If you install the server on the same system as Integration Developer you get a warning about the existing test environment server.

- ▶ Enter the name of the installation directory (Figure A-17). The default is:

C:\Program Files\IBM\WebSphere\ProcServer

C:\WPS602

<==== our choice

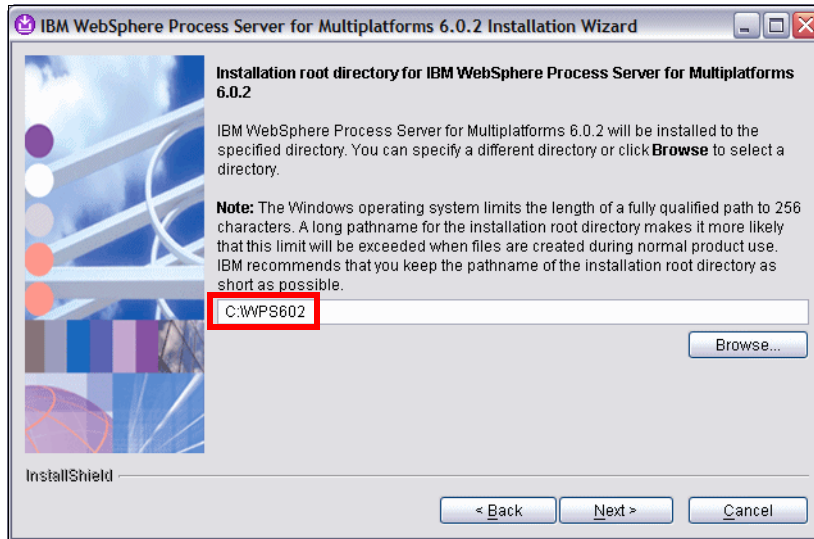


Figure A-17 WebSphere Process Server: Installation Directory

- ▶ Select *Custom installation* (Figure A-18).

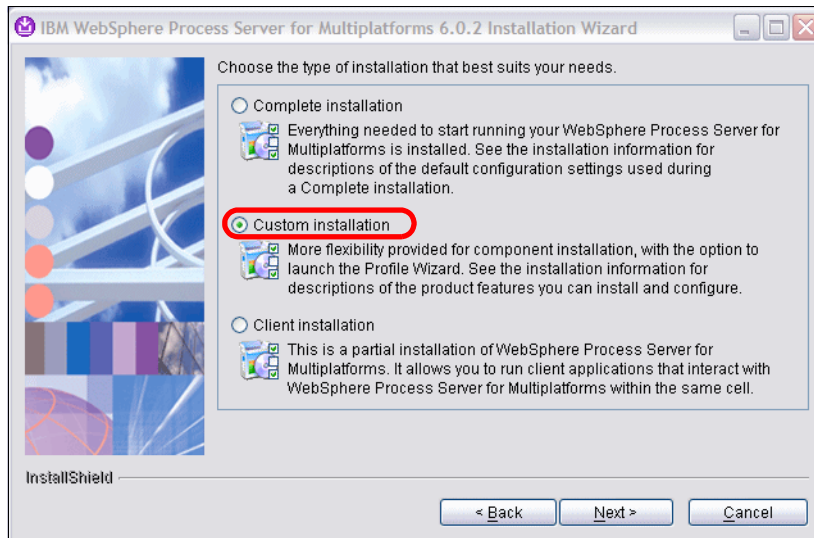


Figure A-18 WebSphere Process Server: Custom installation

- ▶ Keep the default installation options (Figure A-19).

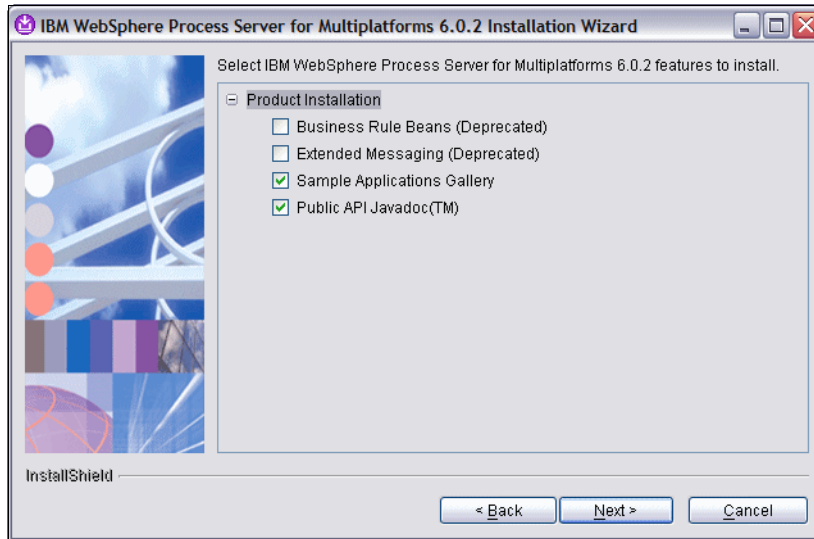


Figure A-19 WebSphere Process Server: Installation options

- ▶ Click *Next* in the summary panel.
- ▶ Wait for the installation to complete.
- ▶ Select *Launch the Profile Wizard* and click *Next* (Figure A-20).

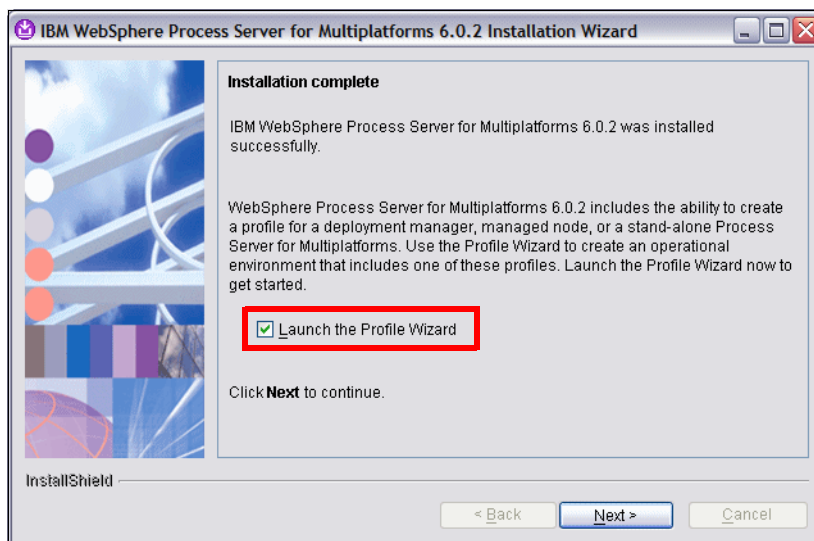


Figure A-20 WebSphere Process Server: Launch Profile Wizard

- ▶ The Profile Wizard starts (Figure A-21). Click *Next*.

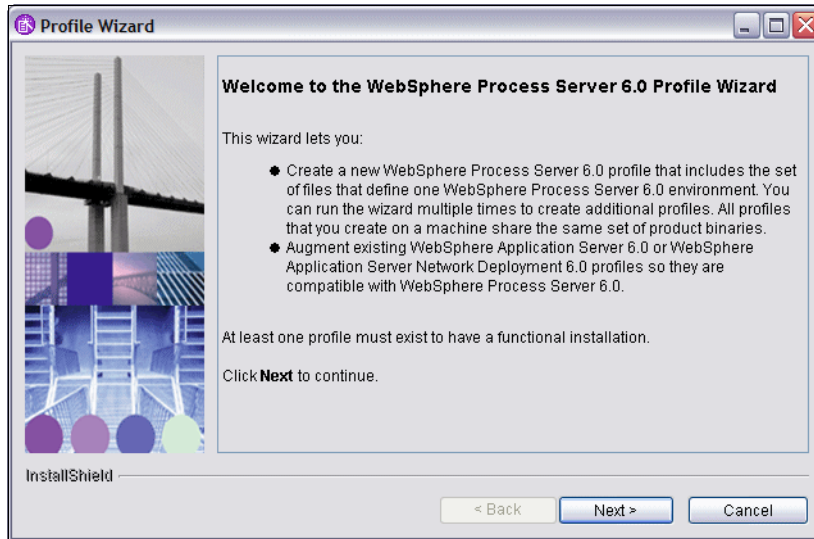


Figure A-21 WebSphere Process Server: Profile Wizard

- ▶ Select *Stand-alone profile* and click *Next* (Figure A-22).

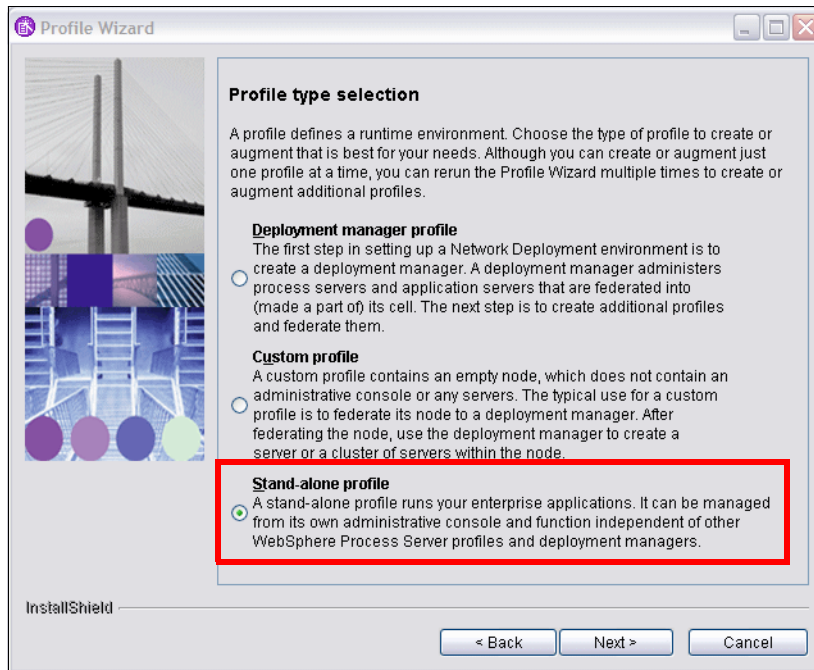


Figure A-22 WebSphere Process Server: Profile selection

- ▶ Enter the name of the profile, the default is ProcSrv01. Click *Next*.
- ▶ Accept the location of the profile directory, for example:
C:\WPS602\profiles\ProcSrv01
- ▶ Enter a node name, for example, WPSNode01, and the host name where the server runs. Click *Next*.
- ▶ Accept the default port numbers. Click *Next*.

Note: If you have other WebSphere servers on the system, the port numbers are increased automatically. If you plan to only run one server at a time, then you can reset all the port numbers to their defaults.

- ▶ Select how to run the server, either as a service or manual.

Note: If you wish to start the service automatically, specify a user with Windows administrative rights.

- ▶ For Service Component Architecture, select *Configure the Service Integration Bus in a secured mode* (Figure A-23).
- ▶ Enter the user ID and password to be used in secure mode. In our example we use wid.
- ▶ Click *Next*.

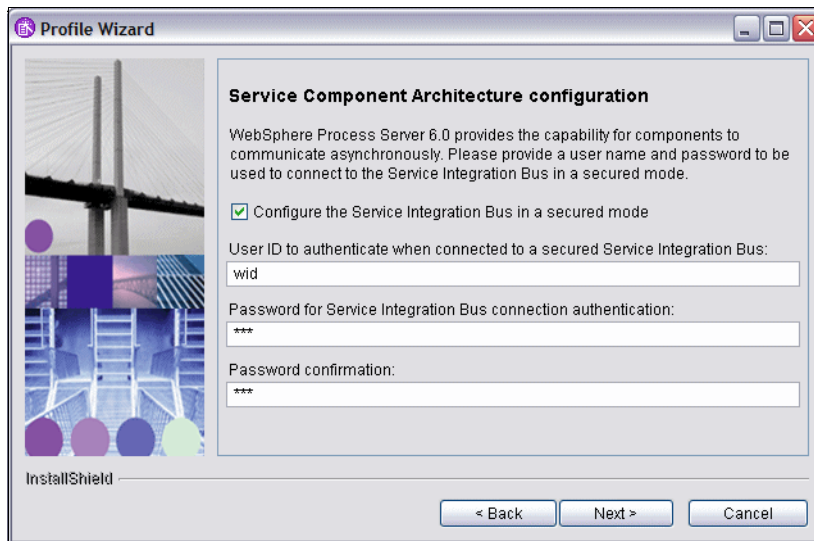


Figure A-23 WebSphere Process Server: Service Component Architecture configuration

- ▶ Enter a valid user ID and password, and database product for the Common Event Infrastructure Configuration (Figure A-24). In our example we use wid.

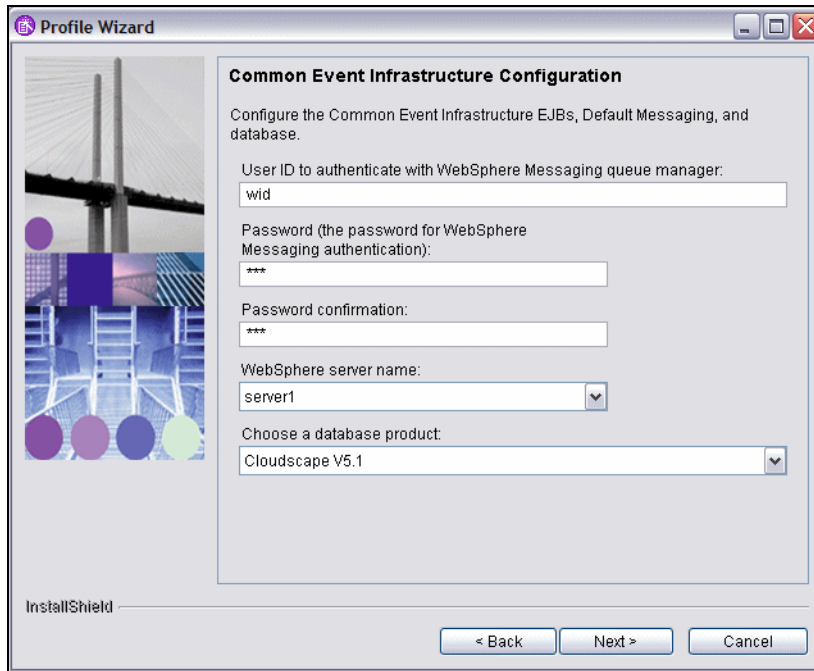


Figure A-24 WebSphere Process Server: CEI configuration

- ▶ Keep the default values for the Business Process Choreographer Configuration. Click *Next*.
- ▶ Click *Next* in the Application Scheduler Configuration panel.
- ▶ Keep the default values on the Database Configuration panel. Click *Next*.
- ▶ Click *Next* in the summary panel.
- ▶ Select *Launch the First Steps console*. From the First Steps console, you can administer, start, and stop the server and run an Installation Verification Test.
- ▶ Click *Finish* to exit the profile wizard.
- ▶ In the First Steps panel, select *Installation verification* (Figure A-25). This starts the server and runs a number of tests to verify the installation of Process Server is functioning correctly.

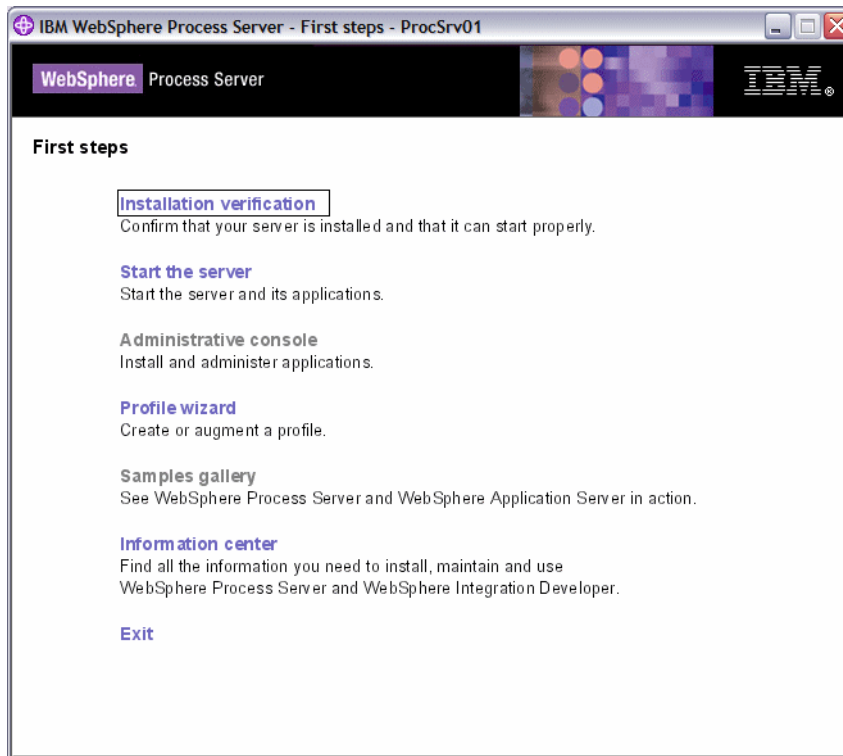


Figure A-25 WebSphere Process Server: First Steps

- Make sure that you see IVT Verification Succeeded (Figure A-26).

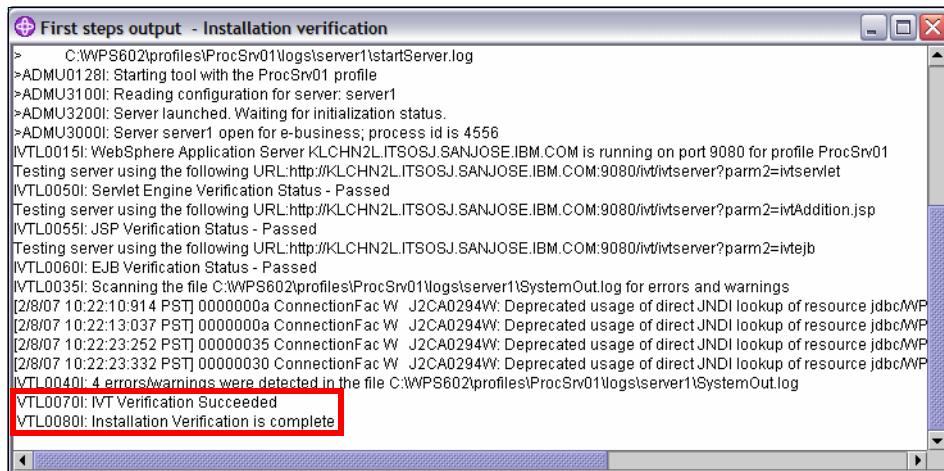


Figure A-26 WebSphere Process Server: Installation Verification

- ▶ Close the Installation Verification panel.
- ▶ Click *Exit* to terminate the First Steps panel.

In the Launchpad select *Additional software installation*. You have the options of installing (Figure A-27):

- ▶ IBM HTTP Server
- ▶ Web Server Plug-Ins
- ▶ Application Clients
- ▶ Application Server Toolkit

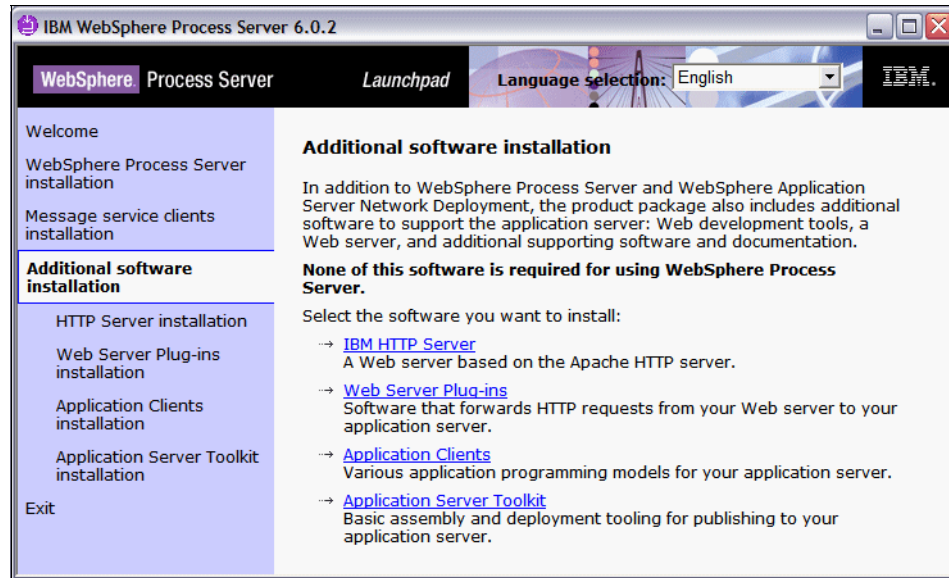


Figure A-27 WebSphere Process Server: Additional software

For the purpose of the ClipsAndTacks scenario, we do not require any of these products.

- ▶ Click *Exit* to terminate the Launchpad.

Interim fixes and fix packs for Process Server

Investigate if interim fixes or fix packs are available and install them:

Tip: You can find the latest list of recommended fixes for WebSphere Process Server at:

<http://www-1.ibm.com/support/docview.wss?rs=2307&uid=swg27006649>

- ▶ Mandatory critical fixes are available for WebSphere Process Server as of December 22, 2006.

These fixes must be applied to the stand-alone process server.

Important: When WebSphere Integration Developer is installed, the fixes included in these mandatory critical fixes are already applied to the Process Server test environment during the installation, with the exception of APAR JR25525. This APAR has to be applied manually to the Process Server test environment under Integration Developer.

- ▶ Interim fix JR25528 is available as of December 29, 2006, and a number of additional interim fixes are available after February 26, 2007.

Controlling the Process Server

You can start the process server by selecting *Start* → *All Programs* → *IBM WebSphere* → *Process Server 6.0* → *Profiles* → *ProcSrv01* → *Start the server*.

From the same menu you can also stop the server, start the administrative console, or start the First Steps menu (Figure A-28).

Tip: You can create desktop short cuts to make it easy to start the server by right-clicking *Start the server* and selecting *Send To* → *Desktop (create shortcut)*. Do the same for *Stop the server* and *Administrative console*.

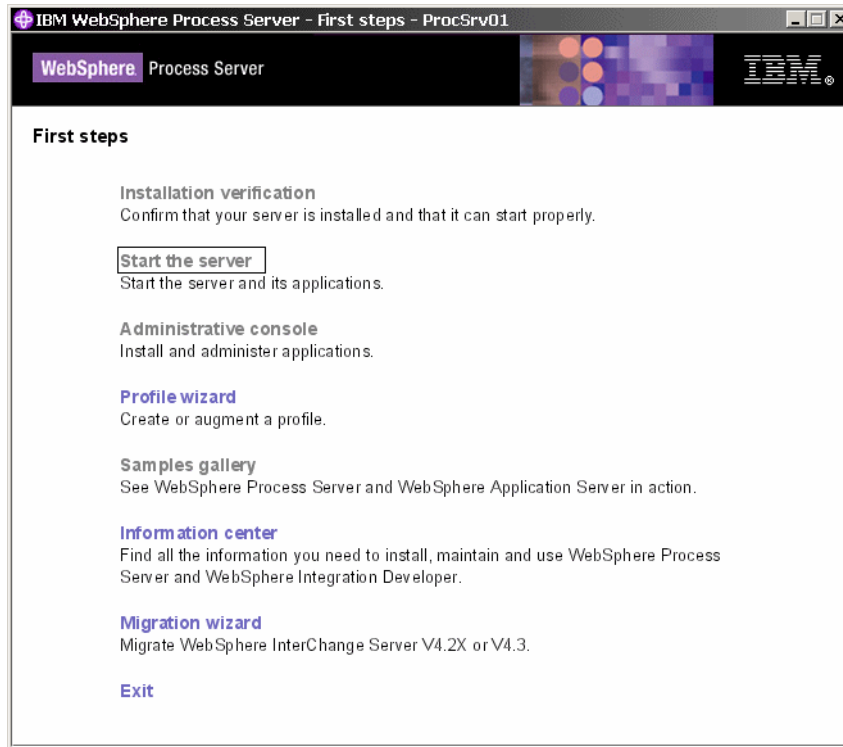


Figure A-28 WebSphere Process Server: First Steps

The first steps menu can be used to verify the installation, start, and stop the server, and start the administrative console.

Creating the database for the business process container

The business process container requires a database. This database is used for internal housekeeping, such as storing state information and business objects for long running process.

For our purpose we use a Cloudscape database. Note that this is not supported in a real production environment.

The default database location is `<WPS-HOME>\profiles\ProcSrv01\databases`.

To create the BPEDB database, follow these steps:

- ▶ Open a command window in the directory:
`<WPS-HOME>\profiles\ProcSrv01\databases`

- ▶ Copy the file `createDatabase.sql` from the `<WPS-HOME>\dbscripts\ProcessChoreographer\Cloudscape` directory to the `databases` directory.

- ▶ Execute the following command to set up the WebSphere environment variables:

```
<WPS-HOME>\bin\setupCmdLine.bat
```

- ▶ Open the file `createDatabase.sql` in WordPad and read the instructions, which tell you to run the command:

```
java -Djava.ext.dirs=%WAS_HOME%\cloudscape\lib  
-Dij.protocol=jdbc:db2j: com.ibm.db2j.tools.ij  
createDatabase.sql
```

Note: If the command returns the following error:

```
'java' is not recognized as an internal or external command, operable  
program or batch file.
```

Then you have to prefix `java` with the path to the Java runtime environment. The Java runtime is located in `<WPS-HOME>\java\bin`.

Tip: If you want to study the output of this command then pipe the output to a text file by appending `> filename` to the command.

- ▶ Executing the command creates the BPEDB database and tables.

The Business Process Choreographer Observer, which is a new component in Version 6.0.2, also requires a database. In our case we will use the BPEDB database to hold the required tables.

Follow these steps to create the tables:

- ▶ Copy the `createSchema_Observer.sql` file from `<WPS-HOME>\dbscripts\ProcessChoreographer\Cloudscape` to the `databases` directory.

- ▶ Open the file `createSchema_Observer.sql` in WordPad and read the instructions, which tells you to run the following command, specifying BPEDB as the database to use:

```
java -Djava.ext.dirs=%WAS_HOME%\cloudscape\lib  
-Dij.protocol=jdbc:db2j:  
-Dij.database=BPEDB  
com.ibm.db2j.tools.ij createSchema_Observer.sql
```

- ▶ Executing the command creates the required tables.

Configuring the business process container

For business process execution, the business process container must be installed using the administrative console or a JACL script.

Start the administrative console and perform these steps:

- ▶ Log in with any user ID—because security is not enabled, this user ID is only used for logging.
- ▶ Expand *Servers* → *Application Servers*.
- ▶ Click *server1*, then expand *Business process container settings* and click *Business process container*.

You should see messages that the business process container is not installed.

- ▶ Under Additional Properties, click *Business process container installation wizard*.
- ▶ Step 1: Database configuration:
 - For JDBC Provider, select *Cloudscape 5.1 (Cloudscape JDBC provider (XA))*. This action fills all the other fields. User ID and password are not required for Cloudscape. Click *Next*.
- ▶ Step 2: JMS configuration (Figure A-29):
 - For JMS user ID and JMS API user ID, enter the user ID and password of the user that starts the process server. In our example we use the user *wid*.
 - For both security roles, enter *Administrators* (this will map to a group in our LDAP server).
 - Click *Next*.

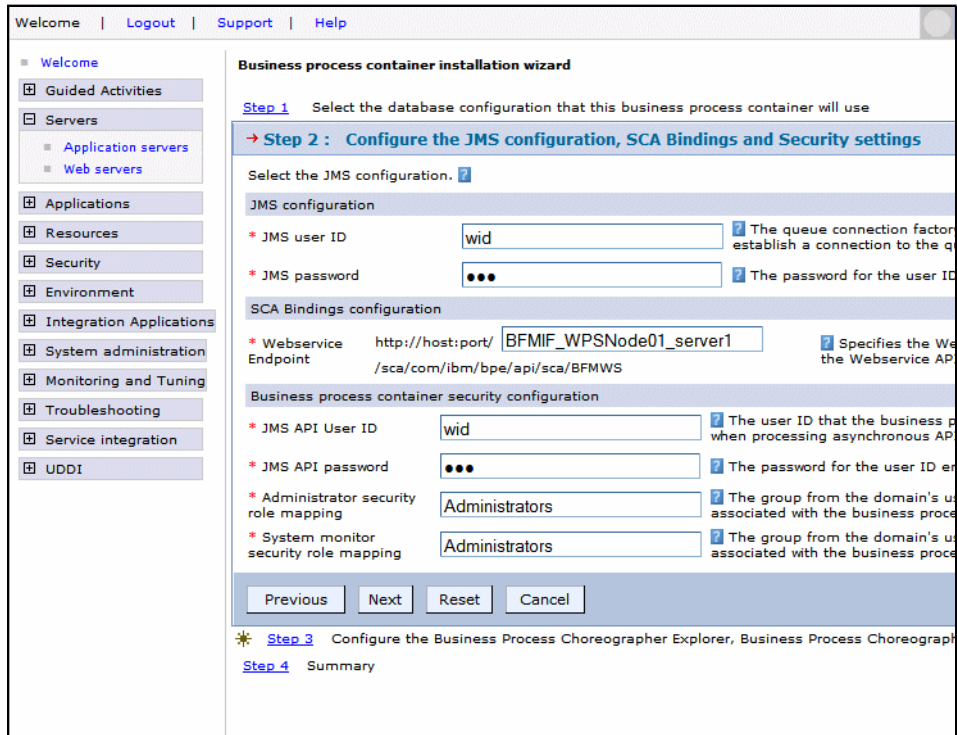


Figure A-29 Configuring the business process container: JMS Configuration

- ▶ Step 3: Configure Business Process Choreographer Explorer and Observer (Figure A-30):
 - Select this check box to install the Business Process Choreographer Explorer (Web client).
 - Enable Common Event Infrastructure logging for all processes running in this container.
 - Select this check box to install the Business Process Choreographer Observer and the Business Process Choreographer Event Collector.
 - Enter the user ID and password for JMS used previously, then click Next.

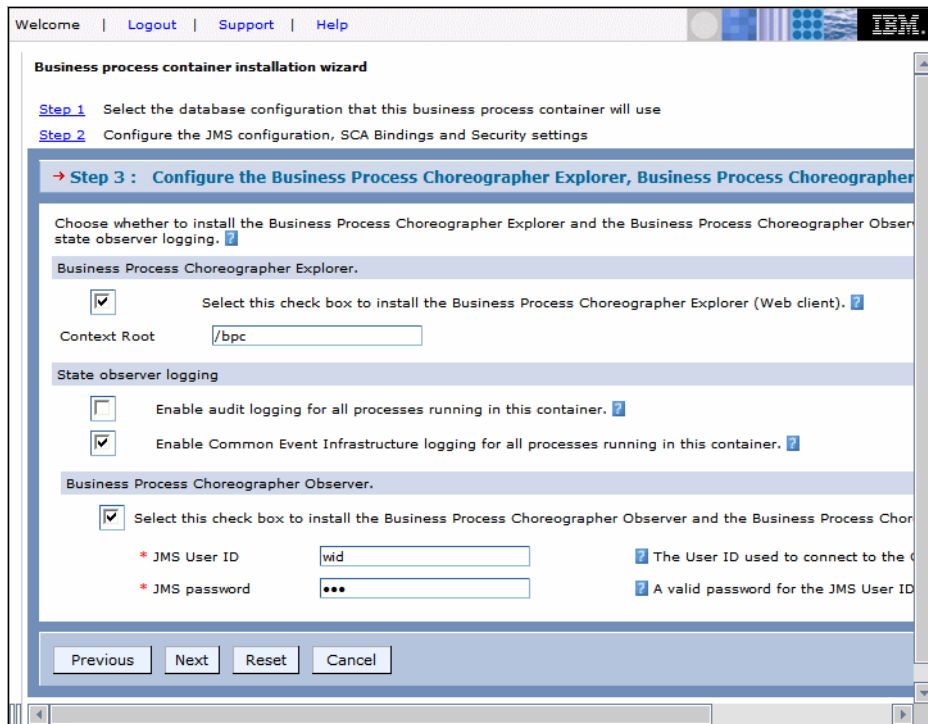


Figure A-30 Configuring the business process container: Components

- ▶ Step 4: Summary:
 - A reminder states: You must create the database and the tables yourself... We have already done this in the previous step. Click *Finish*.
- ▶ The applications are installed. Wait for the messages:
 - Application BPEContainer_WPSNode01_server1 installed successfully.
 - Application BPCEXplorer_WPSNode01_server1 installed successfully.
 - Application BPCObserver_WPSNode01_server1 installed successfully.
 - Application BPCECollector_WPSNode01_server1 installed successfully.
- ▶ Notice the remark: *In order to use human task capabilities, the Human Task Manager must also be configured.*
 - We do this next, so do not save the configuration changes yet.

Note: The Business Process Choreographer Observer is a new component in version 6.0.2. You can use it to create reports on processes and tasks that have completed. You can also use it to view the status of running processes and tasks.

Configuring the human task container

Continue the process and install the human task container:

- ▶ If you still have the result from configuring the business process container open, click *The Human Task Manager configuration is here*. Otherwise, follow these steps:
 - Expand *Servers* → *Application Servers*.
 - Click *server1*, then expand *Human task container settings* and click *Human task container*.
 - You should see messages that the human task container is not installed.
 - Under Additional Properties, click *Human task container installation wizard*.
- ▶ Step 1: JMS configuration:
 - For JMS user ID and Escalation user ID, enter the user ID and password of the user that starts the process server. In our example we use the user `wid`.
 - For both security roles, enter `Administrators` (this will map to a group in our LDAP server).
 - Click *Next*.
- ▶ Step 2: Mail session and logging:
 - Select *Enable Common Event Infrastructure logging* and click *Next*.
- ▶ Step 3: Summary:
 - Click *Finish*.
- ▶ The application is installed. Wait for the message:
`Application TaskContainer_node_server1 installed successfully.`

Save the configuration and restart the server

Click *Save to Master Configuration*, then click *Save*.

Stop and restart the server, for example, using the First Steps menu.

Enabling security for the Process Server

The next step is to enable security on the Process Server. Make sure that you have completed the LDAP installation task in “Installation of IBM Tivoli Directory Server V5.2” on page 612.

- ▶ In the administrative console, expand *Resources* → *Staff plug-in provider*.

- ▶ Click *LDAP Staff Plugin Provider*.
- ▶ Under *Additional Properties* click *Custom properties*.
- ▶ Select the property *ProviderURL* and change the value of the host name from `localhost` to your LDAP Server (Figure A-31):

```
From: ldap://localhost:389
To: ldap://KLCHL2Y.itsosj.sanjose.ibm.com:389
```

ProviderURL	ldap://KLCHL2Y.itsosj.sanjose.ibm.com:389	The provider URL of the LDAP/JNDI server to connect to. Sample: 'ldap://localhost:389'.
-----------------------------	---	---

Figure A-31 Configuring the LDAP Provider URL

- ▶ Expand *Security* → *Global security*
- ▶ Under *Authentication*, expand *Authentication mechanisms* and click *LTPA*.
- ▶ Enter a password and click *Apply*.

Note: Remember the password, as we are going to export the LTPA key from this server and import it to the Monitor Server at a later stage.

Important: If you are configuring security on the Monitor Server and you have already exported the LTPA Key from the Process Server, import the key file here.

- ▶ Enter a file name, `C:\LTPA.key` for example, and select *Export Keys*.
- ▶ Click *OK*.
- ▶ Under *User registries*, click *LDAP*.
- ▶ For *Server user ID*, enter the user ID and password of the user that starts the process server. In our example we use the user `wid`.
- ▶ For *Host*, enter the fully qualified host name of the system where the LDAP Server is located. In our example we use `KLCHL2Y.itsosj.sanjose.ibm.com`.
- ▶ For *Base distinguished name (DN)*, enter the base DN for your LDAP Server. In our example we use `dc=ibm,dc=com`.
- ▶ Leave the remaining default values as is (Figure A-32).

Note: Our LDAP Sever is configured to accept anonymous connections. Bind distinguished name (DN) is where you specify the credentials to connect to your LDAP Server if required.

- ▶ Click *Apply*.

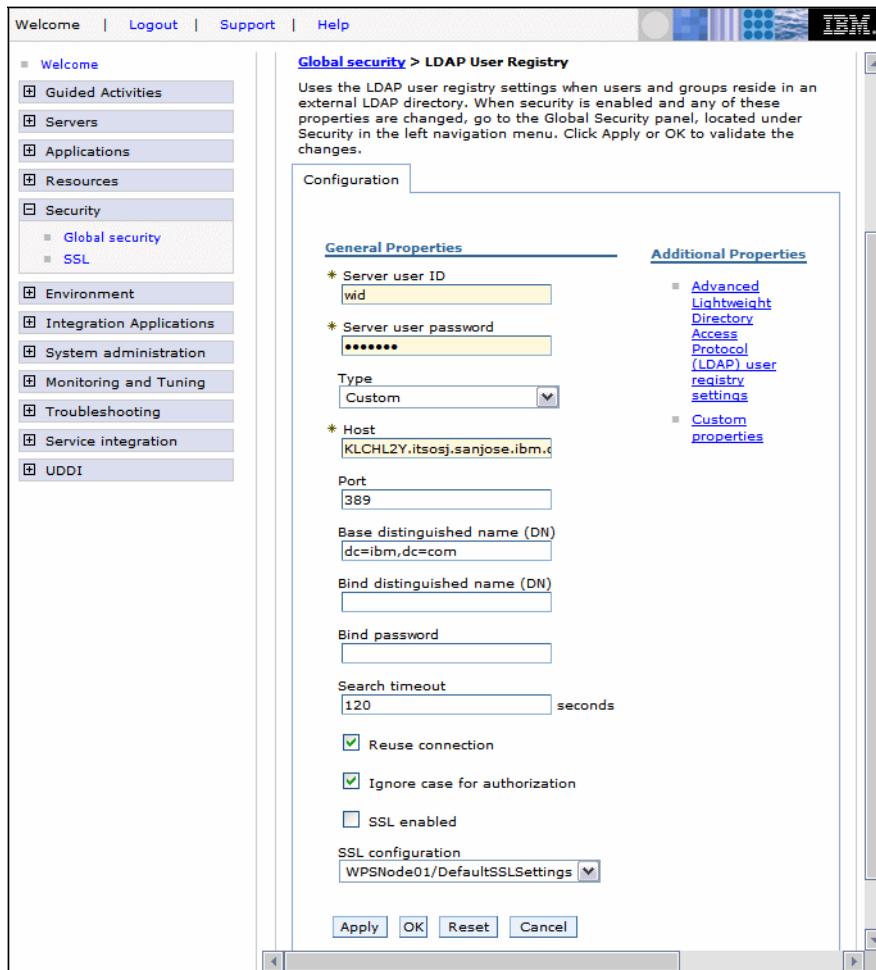


Figure A-32 Enable Security: Specifying LDAP settings

- ▶ Still in the *LDAP User Registry* panel, under *Additional Properties*, click *Advanced Lightweight Directory Access Protocol (LDAP) user registry settings*.
- ▶ We want to be able to look up users by their user ID, so change the setting for *User filter* (Figure A-33):
 - From: (&(uid=%v)(objectclass=ePerson))
 - To: (&(uid=%v)(objectclass=inetOrgPerson))

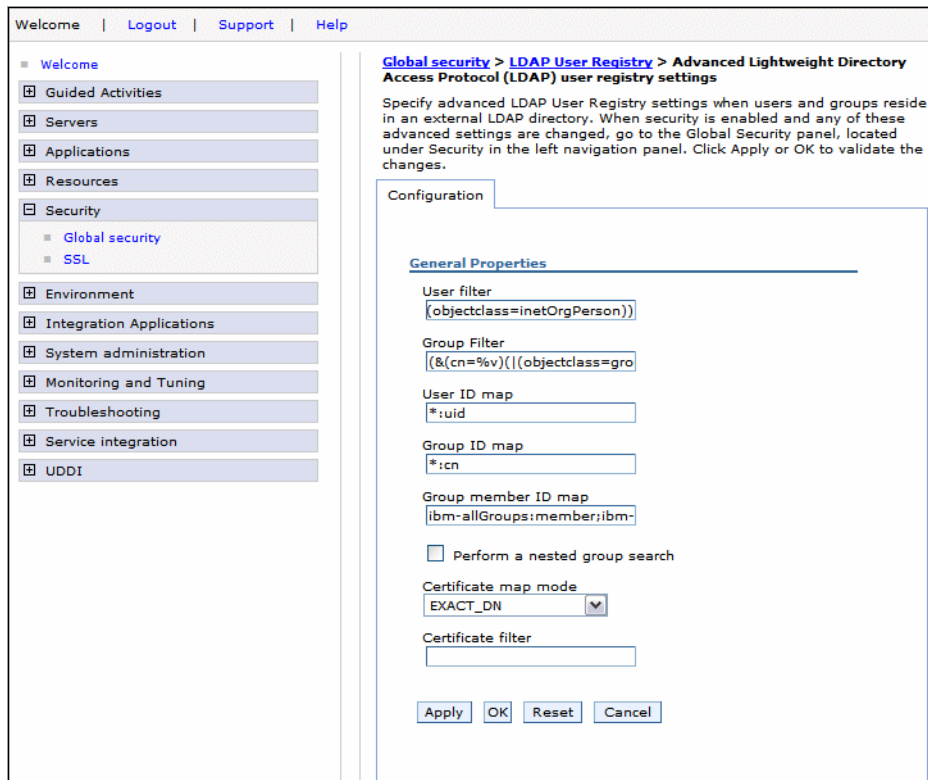


Figure A-33 Enable security: Advanced LDAP user registry settings

- ▶ Click *OK* twice to return to the main *Global security* panel (Figure A-34).
- ▶ Select *Enable global security*. This automatically selects *Enforce Java 2 security*. Clear *Enforce Java 2 security* as we do not require this in our example.
- ▶ For *Active authentication mechanism*, select *Lightweight Third Party Authentication (LTPA)*.
- ▶ For *Active user registry*, select *Lightweight Directory Access Protocol (LDAP) user registry*.
- ▶ Click *OK*.

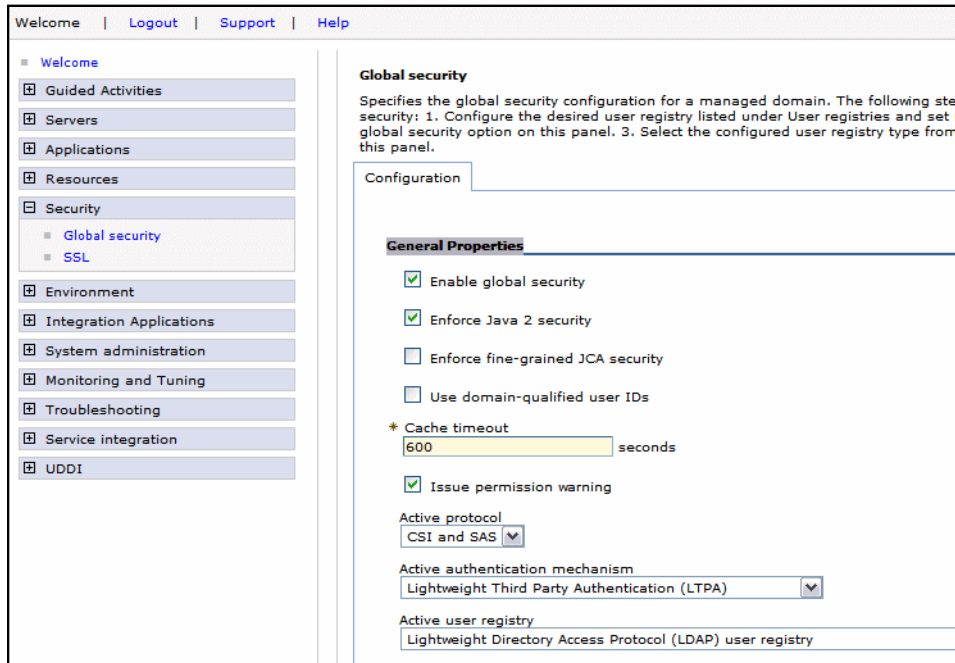


Figure A-34 Enable security: Global security settings

Note: When you click *OK* in the Global security panel, the security settings are validated. If there are any problems with the settings, you will receive an error message at this stage.

- ▶ Save the changes.
- ▶ Restart the server either using the desktop shortcuts created earlier or using the First Steps panel.

Security is now enabled on the process server.

Note: When security is enabled, you are required to log on to the administrative console using a user with proper privileges and a valid password. In our example we can log on with the user `wid`.

You also need to specify a username and password when you stop the server. The command is:

```
stopserver server1 -username wid -password wid
```

If you created a shortcut to stop the server, now would be a good time to update the command—however, you should not store the password in the shortcut for security reasons. If no user ID and/or password is specified, you will be prompted to supply one when you try to stop the server.

Important: There are a number of other essential tasks required to secure a production environment. For information on how to do this, refer to the online Information Center.

Configuring security roles for submitting events

The order process is started using a customer GUI and the user credentials used for accessing the GUI are used to authenticate event submission. Because this user interface is a Web-based application that any Internet user can access, we have to configure the event server to accept that the creator of the event is any unauthenticated user.

To allow any user to submit events, perform the following steps:

- ▶ Open the administrative console and navigate to *Applications* → *Enterprise Applications* → *EventServer*.
- ▶ From the Additional Properties, select *Map security roles to users/groups* (Figure A-35).

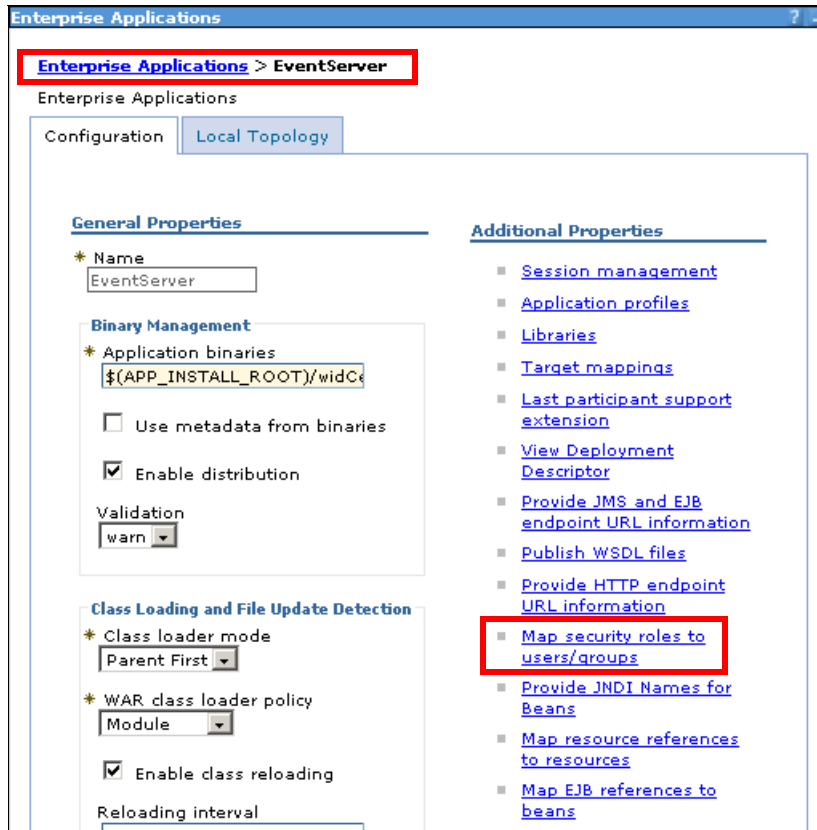


Figure A-35 Mapping security roles for the event server

- ▶ Select *Everyone* for the eventCreator entry (Figure A-36).

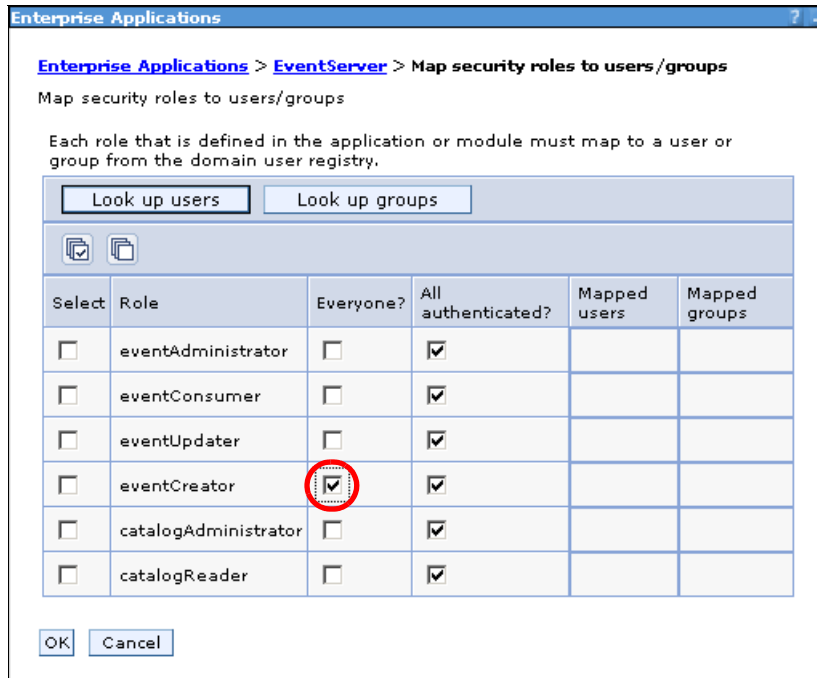


Figure A-36 Specifying that everyone can create events

Verifying the installation

Verify the business process container, business process choreographer explorer and observer are all running:

- ▶ In the Administration console, expand *Applications* → *Enterprise Applications*
- ▶ Verify that the following applications have a status of *running*:

```
BPCECollector_WPSNode01_server1
BPCEExplorer_WPSNode01_server1
BPCEObserver_WPSNode01_server1
BPCEContainer_WPSNode01_server1
```

An application is provided to verify that the business process container and human task container are working. To install the application, follow these steps:

- ▶ In the *Applications* → *Enterprise Applications* panel, click *Install*.
- ▶ Browse the local file system and select the following file:


```
<WPS-HOME>\installableApps\bpcivt.ear
```
- ▶ Click *Next*.
- ▶ Keep the default values in all panels and click *Next* until you get to *Step 9 Summary*.

- ▶ Click *Finish*.
- ▶ Wait for the message:
 - Application BPCIVTApp installed successfully.
- ▶ Click *Save to Master Configuration* followed by *Save* in the next panel.
- ▶ In the *Applications* → *Enterprise Applications* panel, find the BPCIVTApp application and start it.
- ▶ Open an Internet browser and enter this URL:
 - http://localhost:9080/bpcivt
- ▶ The result should be as follows (Figure A-37).

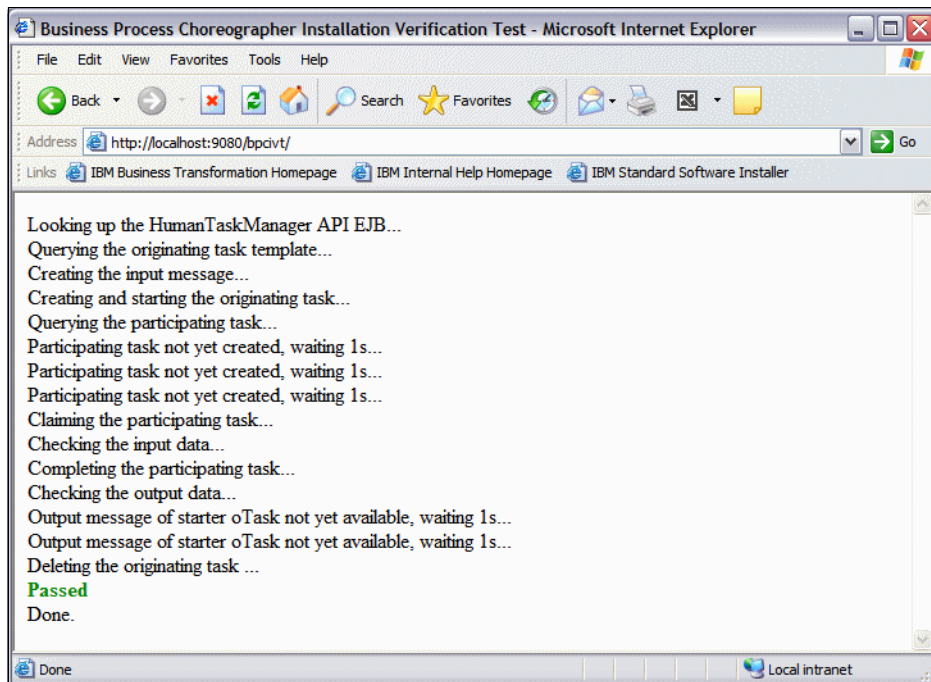


Figure A-37 Business Process Choreographer Installation Verification Tool

Note:

- ▶ If you experience problems while installing the bpcivt application, this is normally caused by difficulties accessing the BPC database.
- ▶ If you experience problems while starting the bpcivt application, this is normally caused by difficulties with the JMS settings or queue managers.

Installation of IBM Tivoli Directory Server V5.2

We use the IBM Tivoli Directory Server as our LDAP server. To install the product, follow these steps:

- ▶ Start the Installation Launchpad using the `\i smp\setup.exe` (Figure A-38).

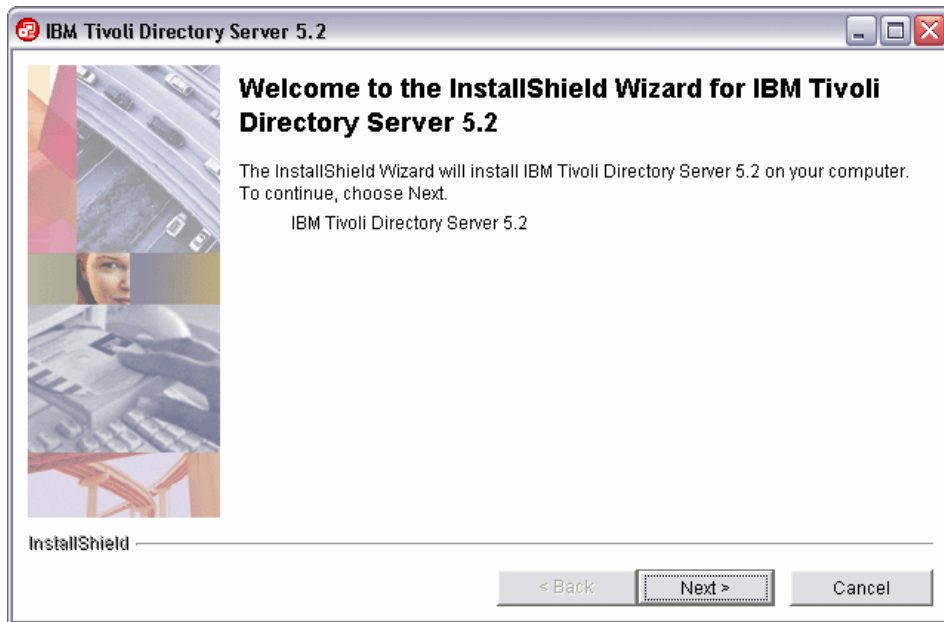


Figure A-38 Tivoli Directory Server: Launchpad

- ▶ Select the language that will be used during the installation process.
- ▶ Accept the license agreement.
- ▶ In the directory window, change the location of the installation of the directory server to `C:\IBM\LDAP`, for example (Figure A-39).



Figure A-39 Tivoli Directory Server: Directory

- ▶ Select the language of the directory server language.
- ▶ Select the components required for the installation (Figure A-40).

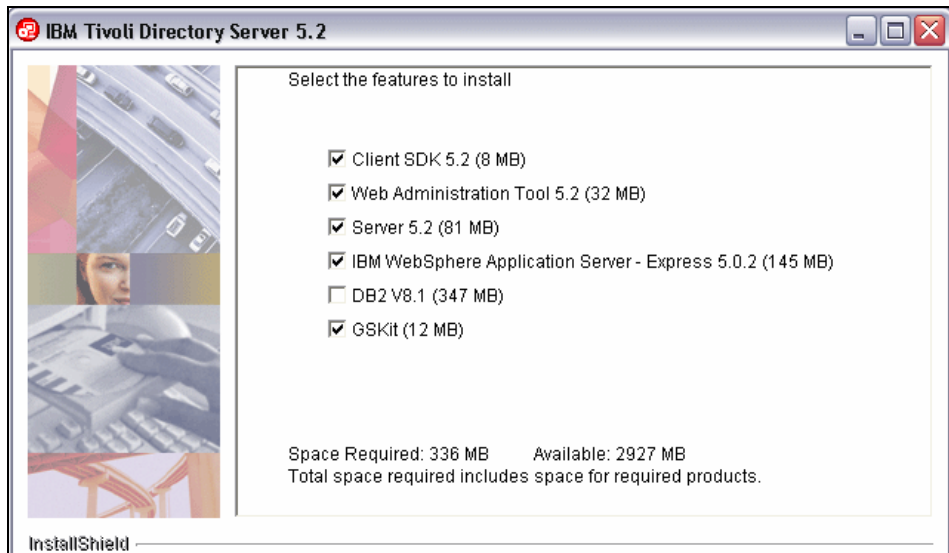


Figure A-40 Tivoli Directory Server: Components

Note: If IBM DB2 exists on the same system as the directory server, then it could be used instead of deploying another DB2 on the system.

- ▶ The summary window appears with all the installation details (Figure A-41).

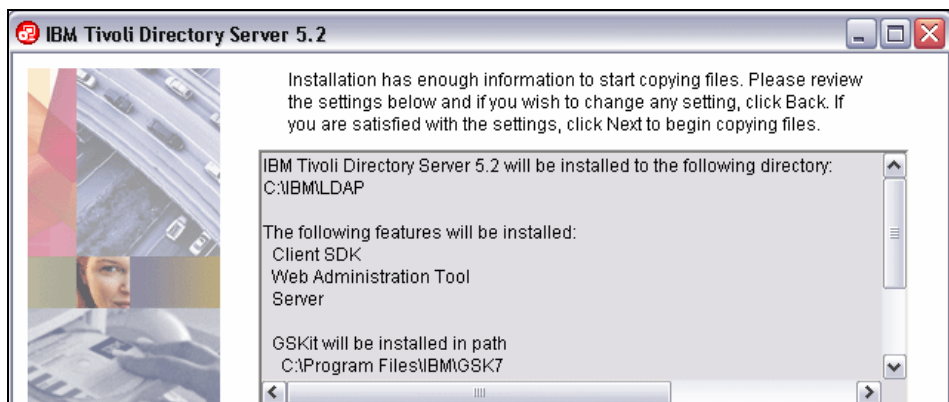


Figure A-41 Tivoli Directory Server: Summary

- ▶ The readme file is displayed.

- ▶ After the installation is performed, the system must be restarted (Figure A-42).

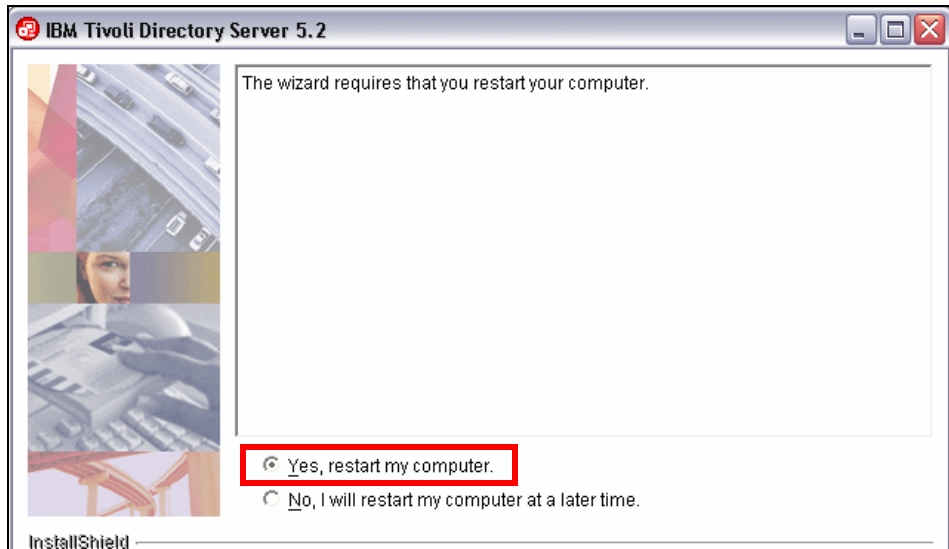


Figure A-42 Tivoli Directory Server: Installation complete

Directory server configuration

To configure the directory server, open the Directory Server Configuration tool using *Start* → *All Programs* → *IBM Tivoli Directory Server 5.2* → *Directory Configuration*:

- ▶ The Welcome screen appears (Figure A-43).

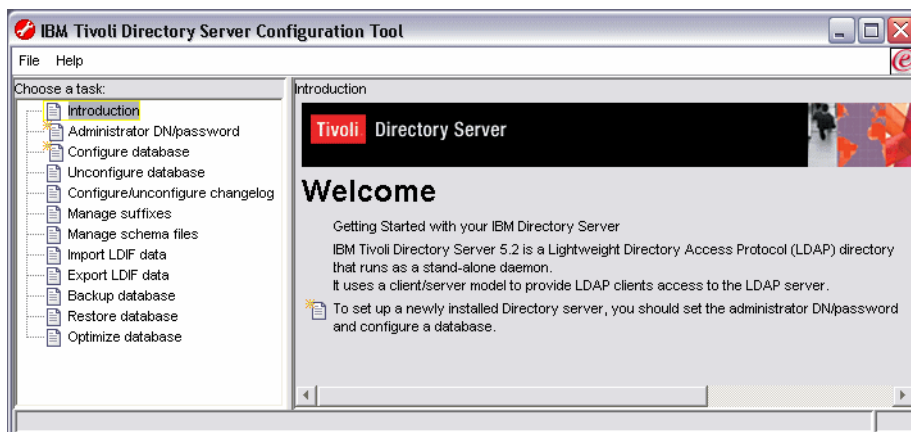


Figure A-43 Directory Server Configuration: Welcome

- ▶ In the left pane, select *Administrator DN/password* (Figure A-44).

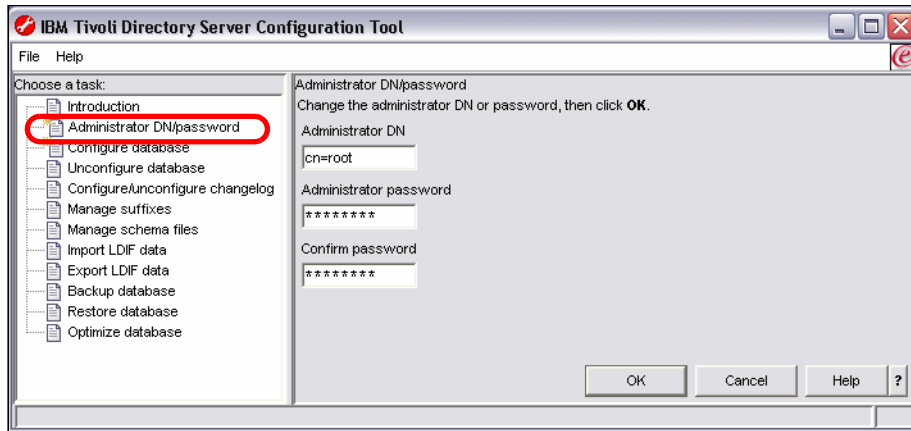


Figure A-44 Directory Server Configuration: Administrator

- ▶ Provide the user name and password of the administrator, for example:
CN=root, bpm13sjc
- ▶ In the left pane, select *Configure Database* and proceed through the dialog (Figure A-45).
 - Enter the user name and password for the database administrator.
 - Give the database a name, for example, LDAP.
 - Select *Create Universal DB2 database (UTF-8/UCS-2)*.
 - Select the drive to install the database on, for example, C.
 - The summary of the database creation configuration appears.
 - Select *Finish* for the database to be created.

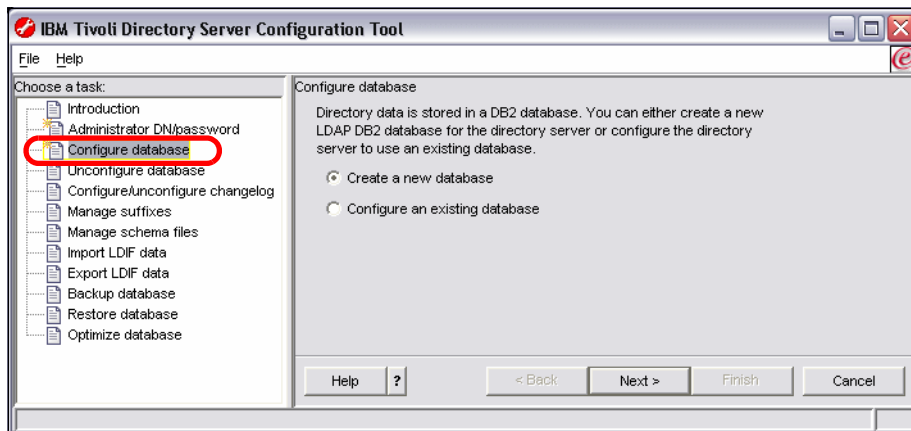


Figure A-45 Directory Server Configuration: Database creation

- ▶ In the left pane, select *Manage suffixes*:
 - In the Suffix DN field, enter `dc=ibm,dc=com`.
 - Click *Add*.
 - Click *OK*.
- ▶ In the left pane, select *Import LDIF data* to import predefined groups and users (Figure A-46):
 - Select the `.ldif` file provided in the sample code:


```
SG247148\sampcode\ldap\ClipsAndTacksUsers.ldif
```
 - Select standard import.
 - Click *Import*.

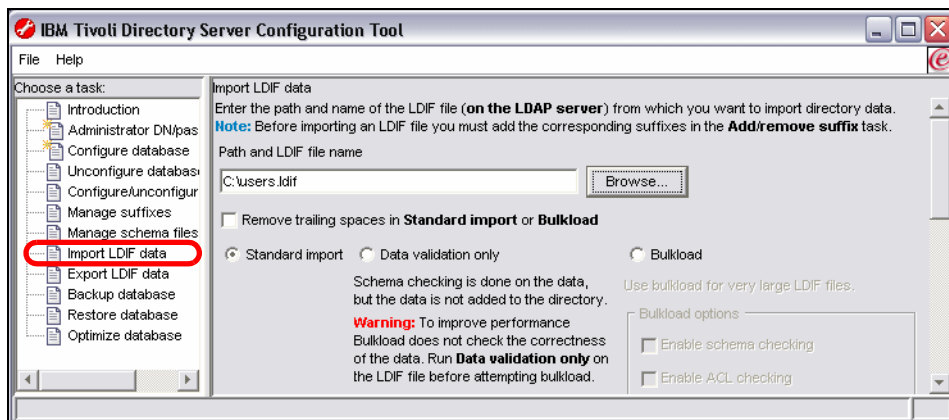


Figure A-46 Directory Server Configuration: Import users

This concludes the installation of the LDAP server.

Starting the LDAP server

Start the LDAP server from the system services (*Administrative Tools* → *Services*) or execute the `ibmsldap` command.

Groups and users

Table A-1 shows the groups and users that we defined for our scenario.

Table A-1 *Groups and users*

Group	User ID	Password	Comment
Administrators	wid	wid	For Process Server administration
administrator	wid	wid	For ClipsAndTacks application administration
	ueli	ueli1	
	db2admin	password	
ordermanager	andre	andre1	For the order manager function in the application
	ueli	ueli1	
shipper	russ	russ1	For the shipper function in the application
	lasrissa	larissa1	
	ueli	ueli1	



B

Installation of WebSphere Business Monitor

This appendix provides the installation procedures for WebSphere Business Monitor V6.0.2

We describe the system and software requirements necessary for installation and the steps involved for a successful installation of WebSphere Business Monitor V6.0.2.

WebSphere Business Monitor prerequisites

This section summarizes the requirements for installation and related components required for WebSphere Business Monitor V6.0.2.

System requirements

Here are the system prerequisites for installing WebSphere Business Monitor for both the Monitor Server and Monitor Dashboard:

- ▶ Windows 2000 Server, service pack 4
- ▶ Windows 2000 Advanced Server, service pack 4
- ▶ Windows Server® 2003 Enterprise Edition service pack 1
- ▶ Windows Server 2003 Standard Edition service pack 1
- ▶ AIX® 5.2, service pack 9
- ▶ AIX 5.3, service pack 4

Software requirements

Figure B-1 shows the software prerequisites for WebSphere Business Monitor.

Prerequisite Software	Monitor Server	Dashboard	Databases
IBM® DB2® Enterprise Server Edition Version 8.2.6	✓	✓	✓
IBM DB2 Cube Views™ Version 8.2.6			✓
IBM WebSphere Process Server Version 6.0.2	✓		
IBM WebSphere Application Server Network Deployment (WAS ND) Version 6.0.2.17		✓	
IBM WebSphere Portal Version 5.1.0.4		✓	
IBM DB2 Alphablox Version 8.4		✓	

Figure B-1 Matrix of prerequisite software requirements

Here are some considerations regarding WebSphere Business Monitor:

- ▶ The WebSphere Business Monitor 6.0.2 installation image should be present in the local system in a directory location without any spaces. Installation from a network drive is not recommended.
- ▶ DB2 Cube Views must be installed on the same system where the DATAMART database resides.
- ▶ If you are installing IBM DB2 Alphablox on a Windows 2003 system, prior to uninstallation, you must set the compatibility level of the file `Uninstall IBM DB2 Alphablox.exe` to the value `Windows XP`.

- ▶ If you already have DB2 Alphablox 8.4 installed, you must apply maintenance to 8.4 using the installation image bundled with WebSphere Business Monitor (8.4 build 122). If you are at a newer level of Alphablox 8.4, do not apply this maintenance.

Important: If you chose the basic install, ensure that your hostname is not monitor. This will cause problems with DB2 later when you create the database with the same name.

Important: Before installing on Windows 2000, you should make the hostname of your system 8 characters or less, if possible. This is important when you remote catalog the Monitor Server DB2 database of this system to your Monitor Dashboard server. Hostnames with more than 8 characters on Windows systems may cause conflicts in the remote catalog of the database system. Also, the WebSphere Application Server cell name can be too long, which could cause problems on Windows systems.

Installation directories

Once you have all of the software prerequisites and Monitor files on your system, you extract the WebSphere Business Monitor Launchpad to the parent directory of the prerequisites. Assuming that you extracted the Launchpad to a folder named CDImage, you should have this directory structure:

CDImage\Alphablox	DB2 Alphablox
CDImage\CUBE	DB2 Cube Views
CDImage\ESE	DB2 Universal Database
CDImage\Portal	WebSphere Portal
CDImage\Portal5104	WebSphere Portal PTF
CDImage\ProcessServer	WebSphere Process Server
CDImage\WAS	WebSphere Application Server

This directory structure is suggested and optimal for a successful run of the Launchpad to install the prerequisites correctly.

If your directory structure containing the installation files for the prerequisites does not match this structure, the launchpad will prompt you to locate the folder where the prerequisite files exist.

Prerequisites

Create a user on your operating system with administrator privileges. We created the user `wid` with a password of `wid`. This is the same user name and password created on the Process Server system.

Important: To avoid problems with different credentials on the Process Server system and the Monitor Server system, we recommend using the same administrative user on both systems.

Note: We work with three different instances of WebSphere Application Server:

- ▶ One instance is on the Process Server system, which is the runtime environment for our business processes. It is located in `C:\IBM\WPS`.
- ▶ Another instance is on the Monitor Server system, which is the runtime environment for the Monitor components. The default install location is in `C:\IBM\Websphere\ProcServer` with additional files located in `C:\IBM\WebSphere\Monitor`.
- ▶ The last instance is also on the Monitor Server system. This instance is the runtime environment for the Dashboard components. The default install location is in `C:\IBM\WAS` with additional files located in `C:\IBM\PortalServer`.

We use one user (`wid`) for administration of all the WebSphere Application Server instances.

Installing the Monitor Server and the Dashboard Server

The Launchpad is used to install the prerequisites and WebSphere Business Monitor for both the Monitor Server and Monitor Dashboard.

Important: The installation procedure in this section only discusses the use of the WebSphere Business Monitor Launchpad to install the prerequisites.

Installation of any of the prerequisites without the use of the Launchpad is not discussed in this book.

Run the Launchpad by starting the batch file `lanchpad.bat`. This starts a command window and the Launchpad opens (Figure B-2).

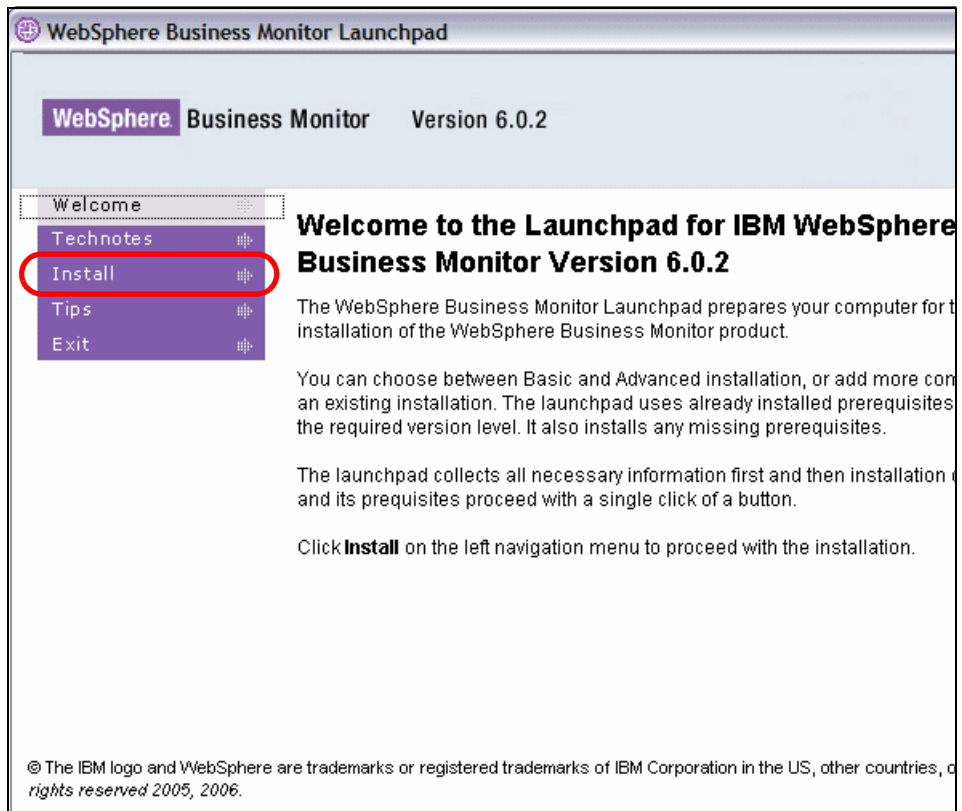


Figure B-2 WebSphere Business Monitor Launchpad

Note: Do not close the command window initially displayed during the installation procedure, as this will abort the WebSphere Business Monitor Launchpad.

License agreement

Accept the License Agreement (Figure B-3).

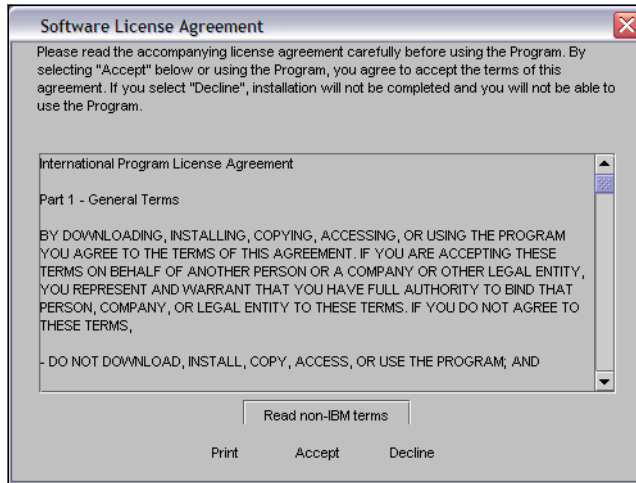


Figure B-3 Monitor Server installation: License Agreement

Installation type

Select *Basic* installation type and click *Next* (Figure B-4). Basic installation is selected for the installation of all the components on one system.

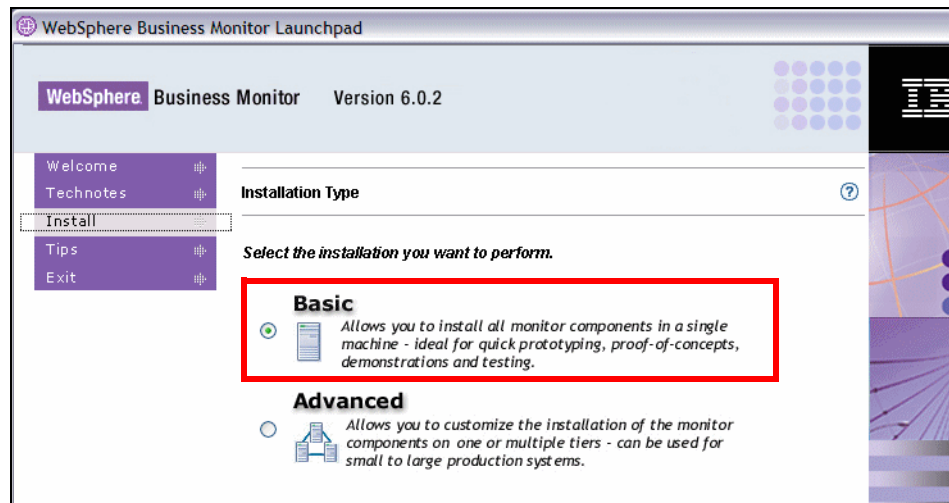


Figure B-4 Monitor Server installation: Basic / Advanced Selection

Installing the Monitor Server (Process Server)

The Monitor Server runs on an instance of Process Server.

Process Server is one of the prerequisites for Monitor Server. The installation launchpad detects an existing installation of Process Server on the system, otherwise Process Server is installed. In both cases a server profile named `wbmonitor` is created.

Specify the directories for the Process Server and the Monitor Server, then click *Next* (Figure B-5).

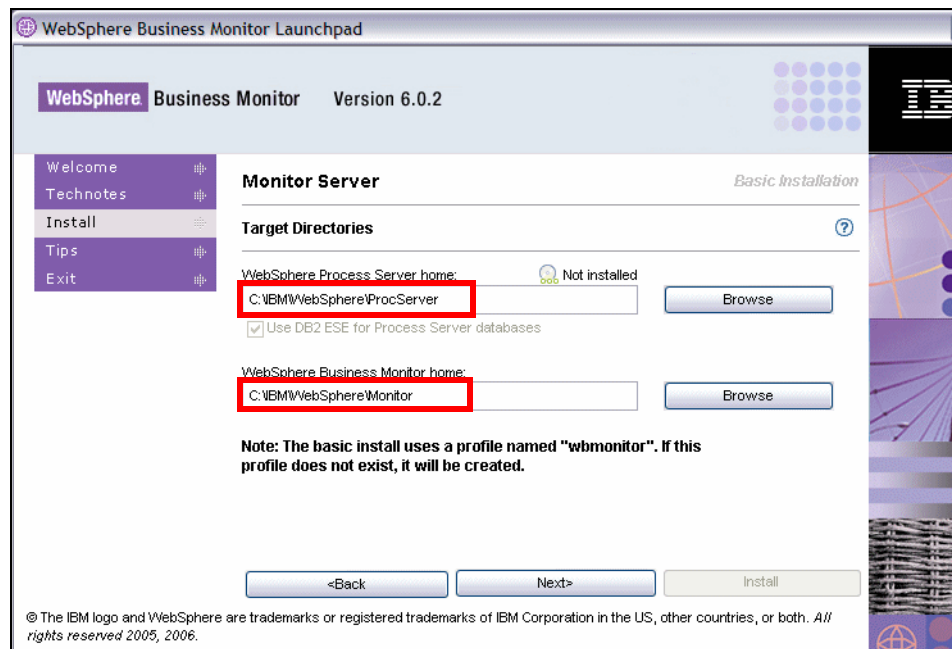


Figure B-5 Monitor Server installation: Directories

DB2

DB2 is one of the prerequisites for the Monitor. The installation launchpad detects an existing installation of DB2 on the system, otherwise DB2 is installed.

Specify the home directory, user ID (`db2admin`), password (`password`), and port for DB2, then click *Next* (Figure B-6).

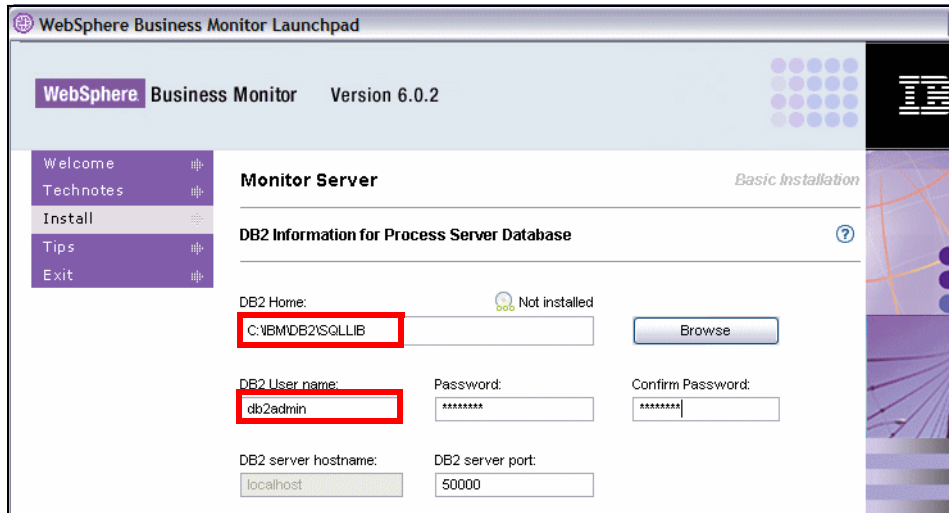


Figure B-6 Monitor Server installation: Database

Monitor Server user

Specify the administrative user for Monitor Server. We used wid with a password of wid, then click *Next* (Figure B-7).

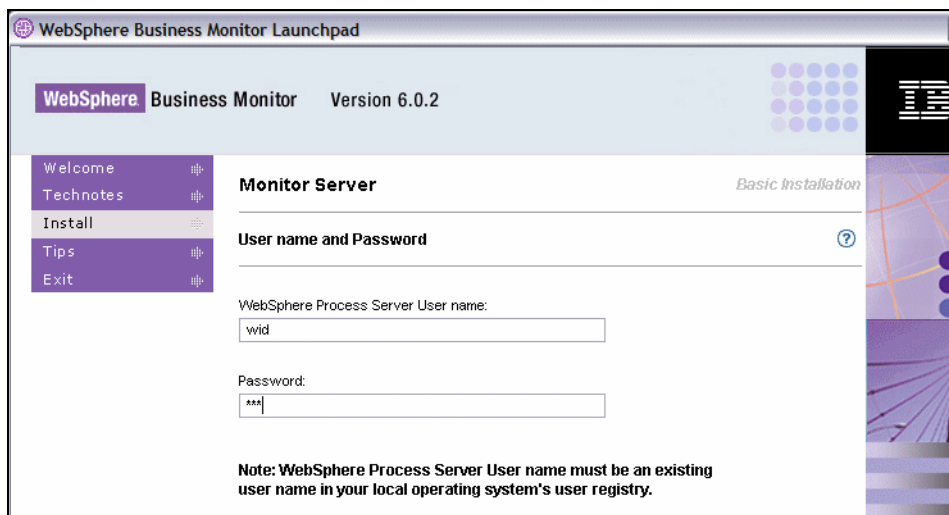


Figure B-7 Monitor Server installation: Process Server username and password

Installing the Dashboard Server

The launchpad detects the presence of the required components, otherwise they will be installed:

- ▶ WebSphere Application Server
- ▶ WebSphere Portal Server
- ▶ DB2 Alphablox
- ▶ WebSphere Business Monitor

Specify the product directories and click *Next* (Figure B-8).

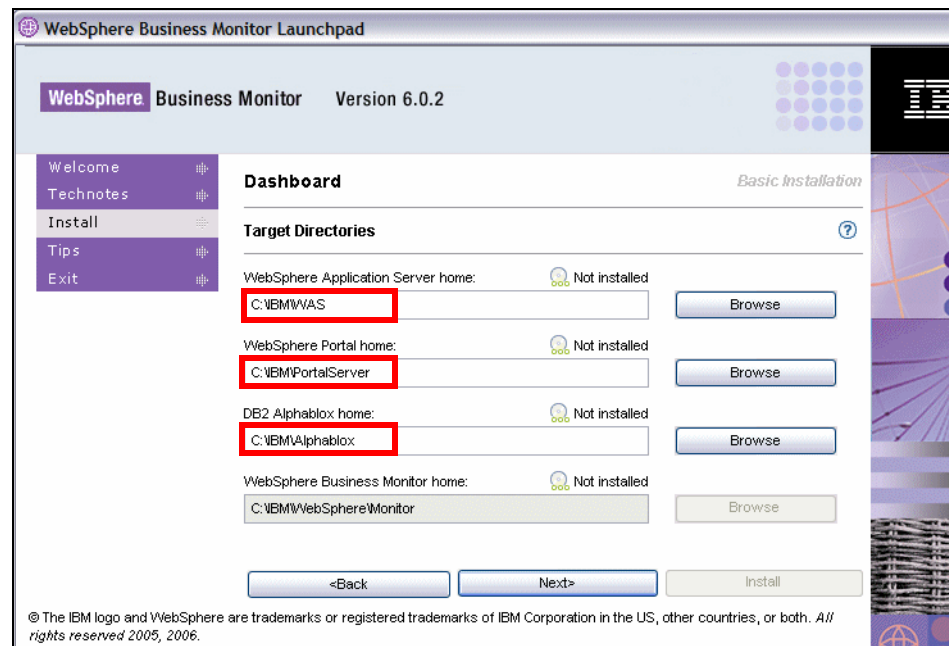


Figure B-8 Monitor Server installation: Dashboard Directories

User name and password

Specify the user name and password for WebSphere Application Server (w i d/w i d), WebSphere Portal Server (wpsadmi n/wpsadmi n), and DB2 Alphablox (admi n/passwor d), then click *Next* (Figure B-9).

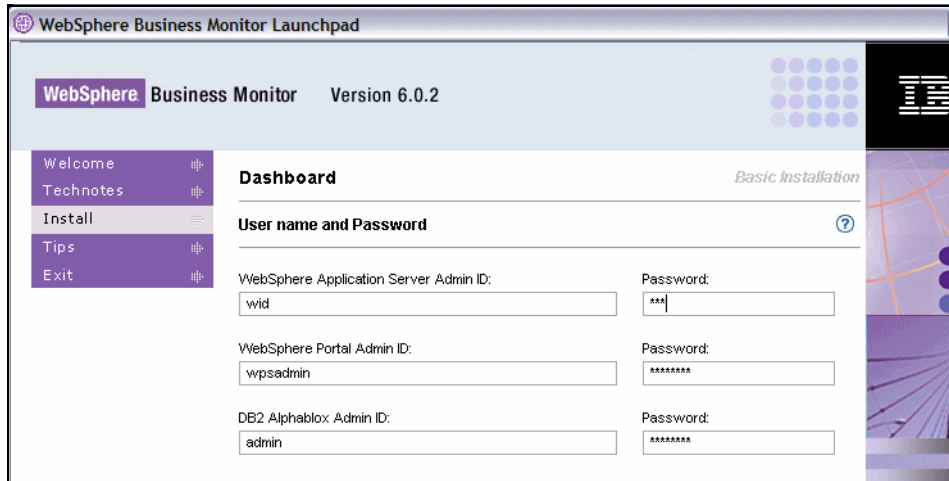


Figure B-9 Monitor Server installation: Dashboard user IDs

Database setup

Specify the user ID and password for the DB2 administrator to run the database DDL scripts to create the MONITOR and DATAMART databases.

To create and manually run the DDL for the two databases, select *Create DDL Scripts only*. Click *Next* (Figure B-10).

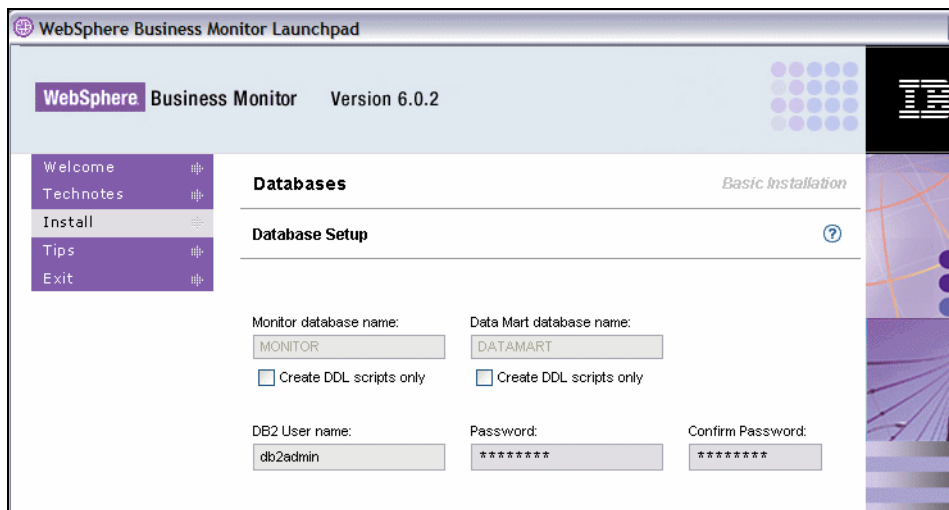


Figure B-10 Monitor Server installation: Database Setup

DB2 backup directory

Specify the backup directories and click *Next* (Figure B-11).

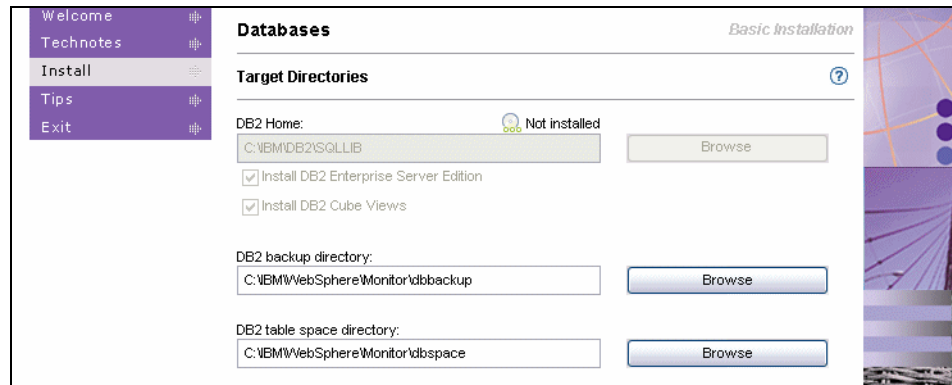


Figure B-11 Monitor Server installation: DB2 Target Directories

Installation

Once all the parameters are specified click *Install* (Figure B-12).

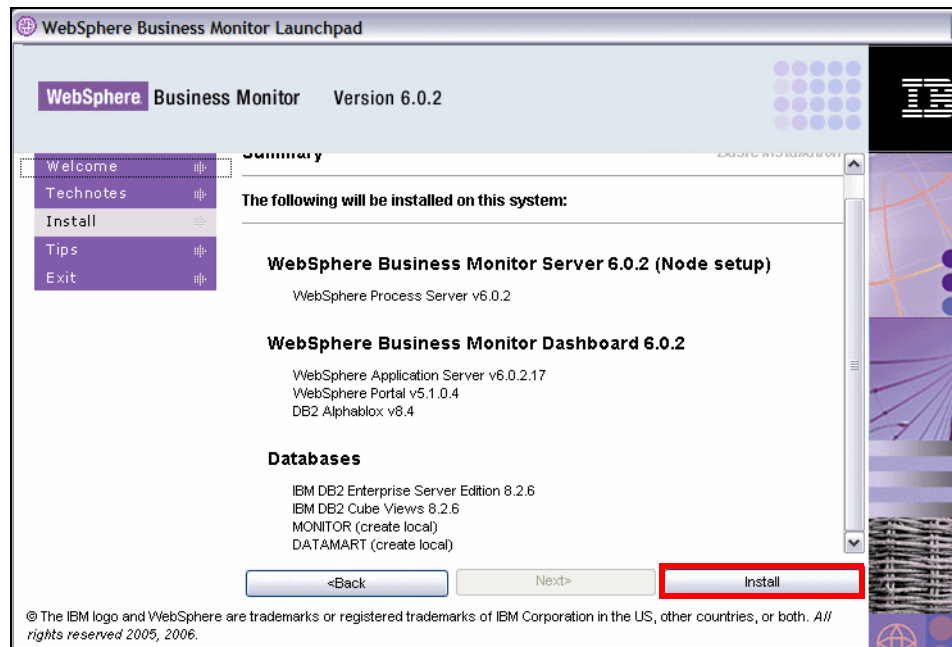


Figure B-12 Monitor Server installation: Install

Security warning

Make sure the firewall products are disabled during the installation (Figure B-13).

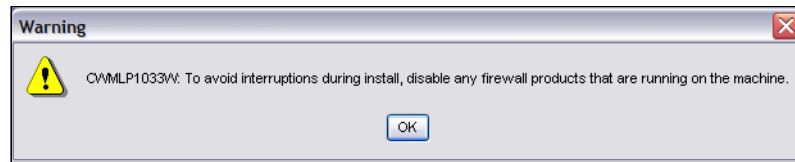


Figure B-13 Monitor Server installation: Firewall

Installation process

The following screens are displaced during the installation process (Figure B-14).

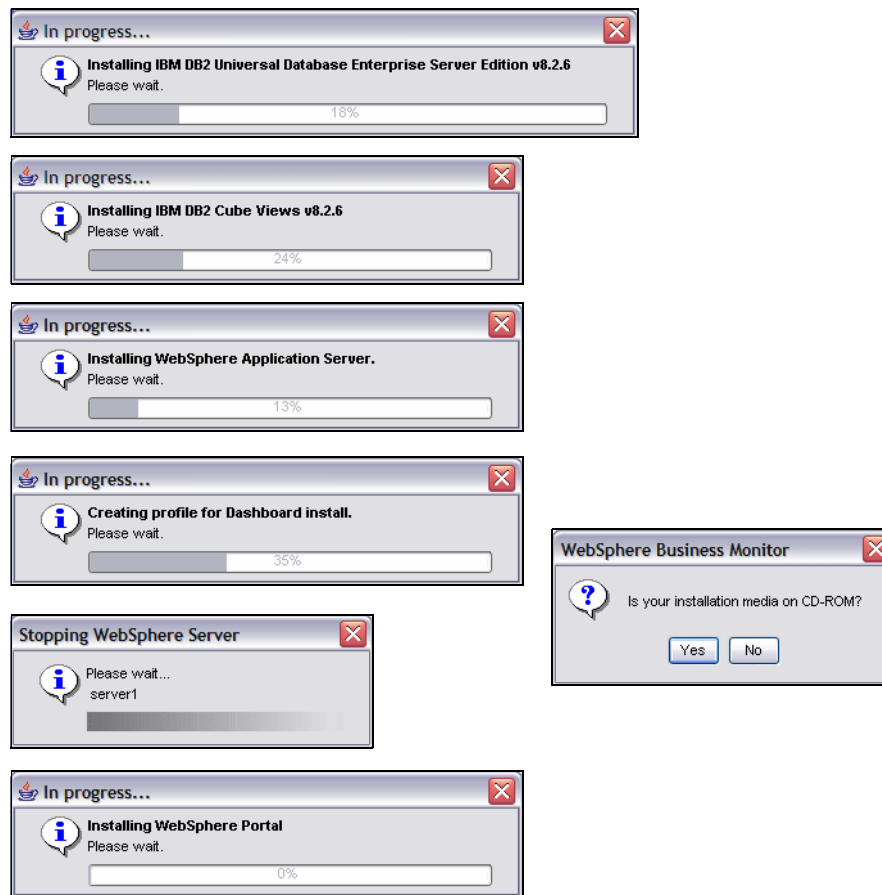


Figure B-14 Monitor Server installation: Progress

Installation status

The installation status is displayed (Figure B-15).

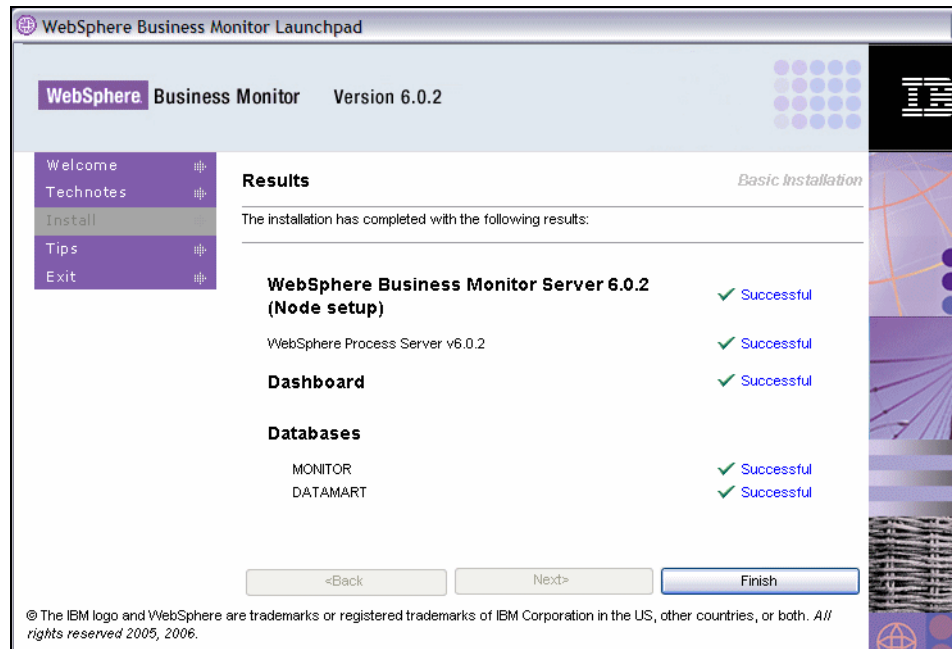


Figure B-15 Monitor Server installation: WebSphere Monitor Install Successful

Starting the Monitor Server

To start the Monitor Server select *Start* → *All Programs* → *IBM WebSphere* → *Process Server 6.0* → *Profiles* → *wbmonitor* → *Start the server*.

Starting the Dashboard Server

To start the Monitor Server select *Start* → *All Programs* → *IBM WebSphere* → *Portal Server v5.1* → *Start the Server*.

Monitor Server service integration bus

The Monitor Server communicates with the WebSphere Process Server that runs the ClipsAndTacks business process on another system. This communication is performed through a service integration bus (SIB) that must be configured.

The SIB is the transport by which the common business events (CBE) are transmitted from the Process Server to the Monitor Server.

SIB configuration for the Monitor Server

Note: The SIB configuration as distributed with the product involves using the administrative console to manually create the SIB configuration. We provide a script in the sample code to simplify the configuration.

First we configure the SIB on the Monitor Server system:

- ▶ Locate the `CrossCellConfiguration` folder in the sample code:

```
SG247148\sampcode\monitor\CrossCellConfiguration
```

- ▶ Edit the `crossCellParameters.tcl` file to reflect your environment for host and WebSphere configuration. You have to enter the cell, node, and hostname (or IP address) for both the Monitor Server and Process Server systems. A sample script is shown in Figure B-16.

```
# -----  
# Monitor Server Parameters  
# -----  
set MONITOR_SERVER_CELL_NAME "KLCHL2YN01C"  
set MONITOR_SERVER_NODE_NAME "Node01"  
set MONITOR_SERVER_SERVER_NAME "server1"  
set MONITOR_SERVER_HOSTNAME "9.43.87.13"  
  
# -----  
# Process Server Parameters  
# -----  
set PROCESS_SERVER_CELL_NAME "KLCHN2LNode01Cell"  
set PROCESS_SERVER_NODE_NAME "WPSNode01"  
set PROCESS_SERVER_SERVER_NAME "server1"  
set PROCESS_SERVER_HOSTNAME "9.43.87.29"
```

Figure B-16 Cross-sell parameters

Tip: To establish the correct values for cell name, node name, and server name, go to <WPS-HOME>\profiles\wbmonitor\config\cells on the Monitor Server and <WPS-HOME>\profiles\ProcSrv01\config\cells on the Process Server. This gives you the cell name. You can drill down the directory structure to determine the values for node name and server name.

- ▶ Save the file and start both the Monitor Server and the Process Server, if they are not started already.

Important: Throughout the entire procedure of configuring the SIB on both the Monitor Server and Process Server systems, ensure that all of the tcl parameters are correct for your system prior to executing the wsadmin scripts.

Once the SIB is configured for a given Application Server, the SIB configuration is stored persistently in the CEI database on the Process Server system.

If you execute the scripts with incorrect parameters, you will have to run the scripts to unconfigure the SIB, which are also provided in the folder.

- ▶ From a command window on the Monitor Server, change the directory to the location of the CrossCellConfiguration folder.
- ▶ Modify the newConfig-Monitor602.bat to reflect the directory structure of your particular setup. If you have followed our suggestions, you need not modify the file.

- ▶ Execute the newConfig-Monitor602.bat command from DOS prompt (Figure B-17).

```

C:\WINDOWS\system32\cmd.exe
C:\IBM\CrossCellConfiguration>newConfig-Monitor602.bat

C:\IBM\CrossCellConfiguration>C:\IBM\WebSphere\ProcServer\profiles\ubmonitor\bin
\wsadmin -f configureMonitorCrossCell.tcl
MASX72091: Connected to process "server1" on node Node01 using SOAP connector;
The type of process is: UnManagedProcess
MONITOR_SERVER_BUS_NAME           = MONITOR.KLCHL2YN01C.Bus
PROCESS_SERVER_BUS_NAME          = MONITOR.KLCHN2LNode01Cell.Bus
SIB_LINK_NAME                     = KLCHL2YN01C.KLCHN2LNode01Cell.link
MONITOR_SERVER_MESSAGING_ENGINE_NAME = Node01.server1-MONITOR.KLCHL2YN01C.Bus
PROCESS_SERVER_MESSAGING_ENGINE_NAME = WPSNode01.server1-MONITOR.KLCHN2LNode01Ce
ll.Bus
PROCESS_SERVER_ENDPOINT_ADDRESS   = 9.43.87.29

## Creating foreign bus: MONITOR.KLCHN2LNode01Cell.Bus...

## Creating service integration bus link routing properties for MONITOR.KLCHN2LN
ode01Cell.Bus...

## Creating destination default for foreign bus: MONITOR.KLCHN2LNode01Cell.Bus..
.

## Creating service integration bus link: KLCHL2YN01C.KLCHN2LNode01Cell.link...

```

Figure B-17 SIB: Create on Monitor Server

- ▶ The SIB is configured on the Monitor Server system.
- ▶ Restart the Monitor Server to allow changes to take effect.

SIB configuration for the Process Server

Configure the SIB on the Process Server system:

- ▶ On the Monitor Server system unzip:
 - <MONITOR-HOME>\install\monsrv\monCrossCell.zip
- ▶ Copy the following three .jar files, extracted from the zip file, to the Process Server system <WPS-HOME>\lib directory:
 - monCrossCellMBean.jar
 - monSPI.jar
 - monSPIImpl.jar
- ▶ Copy the CrossCellConfiguration folder from the Monitor Server system to the Process Server system.
- ▶ From a command window on the Process Server system, change the directory to the location of the CrossCellConfiguration folder.

- ▶ Modify the newConfig-WPS602.bat to reflect the directory structure of your particular setup. If you have followed our suggestions, you need not modify the file.
- ▶ Execute the newConfig-WPS602.bat command from DOS prompt (Figure B-18).

```

C:\WINDOWS\system32\cmd.exe
<C> Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrator>cd C:\IBM\CrossCellConfiguration
C:\IBM\CrossCellConfiguration>newConfig-WPS602.bat
C:\IBM\CrossCellConfiguration>C:\WPS602\profiles\ProcSrv01\bin\wsadmin -f configureCrossCell.tcl
WASX7209I: Connected to process "server1" on node WPSNode01 using SOAP connector
; The type of process is: UnManagedProcess

## Creating SIBus: MONITOR.KLCHN2LNode01Cell.Bus...
## Creating foreign bus: MONITOR.KLCHL2YN01C.Bus...
## Creating service integration bus link routing properties for MONITOR.KLCHL2YN01C.Bus...
## Creating destination default for foreign bus: MONITOR.KLCHL2YN01C.Bus...
## Creating service integration bus link: KLCHL2YN01C.KLCHN2LNode01Cell.link...
## Creating foreign destination: Monitor_Bus_Queue_Destination...
C:\IBM\CrossCellConfiguration>

```

Figure B-18 SIB: Create on Process Server

- ▶ If you have enabled security on the Process Server, you will be prompted for a userid and password. Supply the userid/password which is used for administration of the server. In our case this is the wid user.
- ▶ The SIB is configured on the Process Server system.
- ▶ Restart the Process Server to allow changes to take effect.

Service integration bus post-configuration checkpoint

After configuring the SIB for both the Monitor Server and Process Server, it is important to validate that the SIB is functional between the two hosts:

- ▶ Open the Monitor Server administrative console with URL:
<http://localhost:9062/ibm/console>
- ▶ Navigate to *Service integration* → *Buses* → *MONITOR.xxxxx.Bus* → *Messaging engines* → *Node01.server1-MONITOR.xxxxxC.Bus* → *Service integration bus link* (Figure B-19).

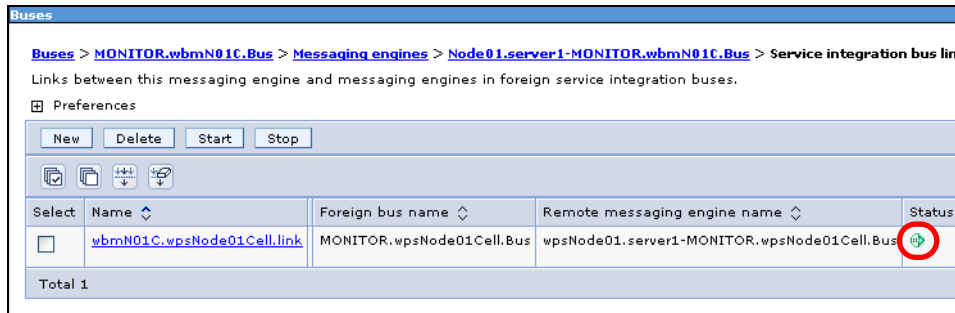


Figure B-19 Monitor SIB status check

- ▶ Look for the green icon in the Status column, which represents the successful creation of the SIB.
- ▶ In the Process Server administrative console, do the same.

Implementing security for the Monitor

This section describes about securing Monitor Server on both Dashboard and Monitor components.

Important: This section assumes that you have installed and configured the LDAP server as described in “Installation of IBM Tivoli Directory Server V5.2” on page 612.

Securing the Monitor Dashboard

This section describes how to secure the Dashboard Portal.

WebSphere Portal provides a number of helper files to assist in carrying out administrative tasks, such as enabling security.

The helper files are located in the <PORTAL-HOME>\config\helpers\ directory. The helper files only contain information relevant for the task at hand, which makes it easy to focus on the properties of importance.

Furthermore, the use of helper files makes it easy to repeat the same administrative task on multiple servers, which need to be configured with the same settings. So, for instance, if you need to enable security for a number of servers, you would customize the appropriate helper file once, copy it to all the servers, and run the same set of commands on each server. Scripting tasks also becomes much easier, as the number of server specific parameters is minimal.

We are configuring the Dashboard to use IBM Tivoli Directory Server, so we modify the file `security_ibm_dir_server.properties`.

We provide a sample `security_ibm_dir_server.properties` file in:

```
SG247148\sampcode\portal
```

Edit the configuration file

Follow these steps to modify the configuration file:

- ▶ Navigate to `C:\IBM\PortalServer\config\helpers`.
- ▶ Make a backup copy of `security_ibm_dir_server.properties`
- ▶ Open the file for editing. Make the following changes matching your environment:

Example 17-1 Editing the `security_ibm_dir_server.properties` file

```
...
```

```
# WasUserId: The user ID for WebSphere Application Server security
WasUserId=uid=wid,cn=users,dc=ibm,dc=com
```

```
# WasPassword: The password for WebSphere Application Server security
WasPassword=wid
```

```
# WpsHostName: The name of the WebSphere Portal host
# Note: a fully-qualified hostname is required to set up security and to
#       communicate with the Portal after security is enabled.
WpsHostName=KLCHL2Y.itsosj.sanjose.ibm.com
```

```
...
```

```
# PortalAdminId: The user ID for the WebSphere Portal Administrator
PortalAdminId=uid=wpsadmin,cn=users,dc=ibm,dc=com
```

```
# PortalAdminIdShort: The short WebSphere Portal admin ID
PortalAdminIdShort=wpsadmin
```

```
# PortalAdminPwd: The password for the WebSphere Portal Administrator
PortalAdminPwd=wpsadmin
```

```
# PortalAdminGroupId: The group ID for the WebSphere Portal Administrator group
PortalAdminGroupId=cn=wpsadmins,cn=groups,dc=ibm,dc=com
```

```
# PortalAdminGroupIdShort: The WebSphere Portal admin group ID
PortalAdminGroupIdShort=wpsadmins
```

```
...

# LTPAPassword: Specifies the password to encrypt and decrypt the LTPA keys.
LTPAPassword=bpm13sjc

...

# SSODomainName: Specifies the domain name (.ibm.com, for example) for all
#                 Single Sign-on hosts.
SSODomainName=

...

# LDAPHostName: The LDAP server hostname
LDAPHostName=KLCHL2Y.itsosj.sanjose.ibm.com

...

# LDAPAdminUid: The LDAP administrator ID
LDAPAdminUid=cn=root

# LDAPAdminPwd: The LDAP administrator password
LDAPAdminPwd=bpm13sjc

...

#LDAPBindID: The user ID for LDAP Bind authentication
LDAPBindID=uid=wpsbind,cn=users,dc=ibm,dc=com

#LDAPBindPassword: The password for LDAP Bind authentication
LDAPBindPassword=wpsbind

...

# LDAPSuffix: The LDAP suffix appropriate for your LDAP server
LDAPSuffix=dc=ibm,dc=com

...
```

Enable security

To enable security, we need to import the new values into `wpconfig.properties`, validate the settings, and finally execute the command to enable security:

- ▶ Open a command prompt at `C:\IBM\PortalServer\config`.

- ▶ To import the new values into `wpsconfig.properties`, execute the command:


```
WPSconfig
-DparentProperties="config\helpers\security_ibm_dir_server.properties"
-DsaveParentProperties=true
```
- ▶ To validate the settings, execute this command:


```
WPSconfig validate-ldap
```
- ▶ Wait for the message:


```
action-validate-ldap-content-admin-group:

BUILD SUCCESSFUL
Total time: 13 seconds
```
- ▶ Enable security by executing this command:


```
WPSconfig enable-security-ldap
```
- ▶ Wait for a successful configuration message:


```
action-start-node-manager:
[logmsg] 2007.03.13 14:27:43.593 enable-security-ldap
[logmsg] EJPCA3153I: Security configured with LDAP

BUILD SUCCESSFUL
Total time: 14 minutes 11 seconds
```
- ▶ Stop and start the Dashboard Server (see “Starting the Dashboard Server” on page 631).

Securing the Monitor Server

To secure the Monitor Server, we have to configure the Process Server on which the Monitor Server runs.

Follow the instructions provided in “Enabling security for the Process Server” on page 603.

Note: Use the same user ID to configure the Monitor Server as you used for the Process Server. In our example we use `wid`.

Set J2C authentication aliases

Security was not enabled when we installed the Monitor components, so we have to modify two of the authentication aliases.

After securing the Monitor Server, follow these steps:

- ▶ Open the administrative console for the Monitor Server.
- ▶ Navigate to *Global Security* → *Authentication* → *JAAS Configuration* → *J2C Authentication data*.
- ▶ Set the MonitorAlphabloxAlias user ID and password. This value has to be set to the user ID and password you specified as the Alphablox administrative user in “User name and password” on page 627. In our case that was admin/password.
- ▶ Set the MonitorQueueConnectionFactoryAuth user ID. This value has to be set to the use ID and password you specified as the Monitor administrative user in “Monitor Server user” on page 626. In our case that was wid/wid (Figure B-20).

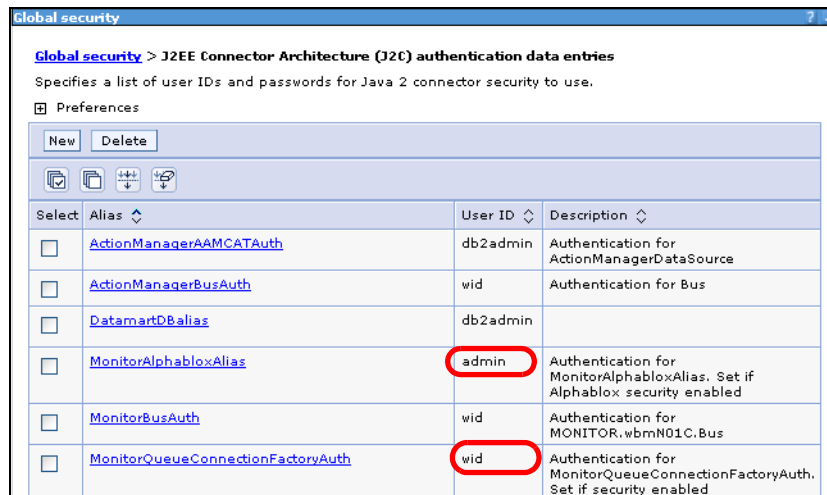


Figure B-20 Setting J2C Authentication Aliases

- ▶ Stop and start the Monitor Server (see “Starting the Monitor Server” on page 631).

Import LTPA keys

The LTPA keys are important in securing the communication between the Process Server, Monitor Server, and Dashboard components. Exchanging the keys between the three WebSphere Application Server instances ensures that the communication can be encrypted and decrypted correctly.

When we secured the Process Server, we exported the LTPA keys to a file on that system. Bring that file across to the Monitor Server system.

Importing the LTPA keys to the Monitor Server

Follow these steps to import the keys to the Monitor Server.

- ▶ In the Admin Console for Monitor Server, navigate to *Security* → *Global Security*.
- ▶ In the *Authentication* section, expand *Authentication mechanisms* and select *LTPA*.
- ▶ Enter the password, the filename, and the location of the LTPA Key file and select *Import Keys*.
- ▶ Save the configuration changes.

Importing the LTPA keys to the Dashboard Server

In order to import the keys to the Dashboard Server, we have to start the administrative console on the WebSphere Application Server instance the Dashboard executes on. The default install location is C:\IBM\WAS. The console is deployed on server1, which is not started automatically.

Follow these steps to import the keys to the Dashboard Server:

- ▶ Open a command prompt.
- ▶ Change the directory to C:\IBM\WAS\bin.
- ▶ Issue the following command:

```
startserver server1
```
- ▶ When the command completes, open a Web browser and go to:

```
http://localhost:9060/admin
```
- ▶ In the Admin Console for Monitor Server, navigate to *Security* → *Global Security*.
- ▶ In the *Authentication* section, expand *Authentication mechanisms* and select *LTPA*.
- ▶ Enter the password, the filename, and the location of the LTPA Key file and select *Import Keys*.
- ▶ Save the configuration changes.

DB2 AlphaBlox

Once security is enabled on the server, DB2 AlphaBlox might become non-functional and have to be reinstalled.

To overcome this problem, you need to reinstall DB2 AlphaBlox:

- ▶ In the Windows Control Panel, select *Add or Remove Programs*.
- ▶ Remove IBM DB2 Alphablox.
- ▶ Go to CDImage\Alphablox\Windows.
- ▶ Execute `install.exe`.
- ▶ Accept the License Agreement. Click *Next*.
- ▶ Optionally change the installation directory to match the original directory. Click *Next*.
- ▶ Leave *Typical* selected. Click *Next*.
- ▶ Click *Next*.
- ▶ Verify that you do not have any WebSphere processes running. Click *Next*.
- ▶ If required, change the location of you WebSphere installation. Click *Next*.
- ▶ Leave *WebSphere_Portal* selected. Click *Next*.
- ▶ Specify the WebSphere Administrator user ID and password. In our case it is `wpsbind/wpsbind`. Click *Next*.
- ▶ Click *Next*.
- ▶ Optionally change the location of the J2SE™ SDK. Click *Next*.
- ▶ Leave JDBC Drivers settings at default values. Click *Next*.
- ▶ Leave Repository settings at default values. Click *Next*.
- ▶ Click *Install*.
- ▶ When the installation completes, click *Done*.
- ▶ Close the installation log, which automatically opens in a browser window.
- ▶ Start the Dashboard server.
- ▶ Open the Alphablox Admin console at the following URL in Internet Explorer:
<http://localhost:9081/AlphabloxAdmin>
- ▶ Log in with the WebSphere Administrator, in our case, `wid/wid`.
- ▶ Navigate to *ADMINISTRATION* → *Data Sources*
- ▶ Click *Create* (Figure B-21).

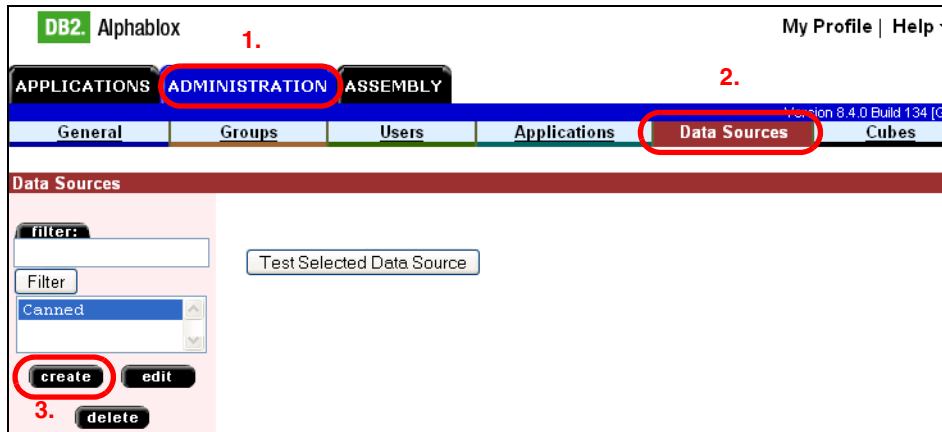


Figure B-21 DB2 Alphablox Administration Console

- Specify the properties as stated in Table B-1.

Table B-1 DATAMART data source values

Property	Value
Data Source Name	DATAMART
Adapter	IBM DB2 JDBC Type 4 Driver
Server Name	localhost
Port Number	50000
Database Name	DATAMART
Default Username	db2admin
Default Password	db2admin
Use DB2 Alphablox Username and Password	No
Maximum Rows	10000
Maximum Columns	1000
JDBC Tracing Enable	No

- Click *Save*.
- Click *Test Selected Data Source* and verify that the connection was successful.

- ▶ Create the data source for DATAMART_CUBE as stated in Table B-2.

Table B-2 DATAMART_CUBE data source values

Property	Value
Data Source Name	DATAMART_CUBE
Adapter	Alphablox Cube Server Adapter
Maximum Rows	1000
Maximum Columns	1000

- ▶ Click *Save*.
- ▶ Click *Test Selected Data Source* and verify that the connection was successful.
- ▶ Close the browser window.

Fix pack installation

Make sure that you update the Monitor Server with the latest fix pack.

This section shows how to install Fix Pack1.

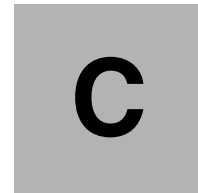
Note: This process is not the same for any new fix pack released for Monitor Server.

- ▶ Download and unzip the fix pack 1 for Monitor Server to C:\temp\.
- ▶ Locate the WebSphere Business Monitor Dashboard war file, dashboardjsr168.war. In our case it is located in C:\IBM\WebSphere\PortalServer\InstallableApps.
- ▶ Make a backup copy of the file.
- ▶ Start the Portal Server on the WebSphere Business Monitor Dashboard Client system.
- ▶ Start the Portal administration using a Web browser with the URL:
<http://localhost:9081/wps/portal>
- ▶ Click *Login* in the upper right corner of the window.
- ▶ Type in the admin user ID and password to log into portal. In our example, this is wpsadmin/wpsadmin.
- ▶ Select *Administration*.

- ▶ In the left column, select *Portlet Management*.
- ▶ Select *Web Modules*.
- ▶ Find `dashboardjsr168.war` by typing `dashboard` into the Search text area. The type of search should be set to *File name starts with*. Click *Search*. The `dashboardjsr168.war` file should appear in the list.
- ▶ Beside the `dashboardjsr168.war` file, click *Update Web Module* (icon). Use the hover-text to help with this.
- ▶ Browse to the newly extracted `dashboardjsr168.war` file. We extracted the file to `C:\temp\dashboardjsr168.war`.
- ▶ Select *Next*, and a list of portlets found within the `dashboardjsr168.war` file appears.
- ▶ Select *Finish* and the WAR file should be updated.
- ▶ Restart the Portal Server.

Tip: You can find the latest list of recommended fixes for WebSphere Business Monitor at:

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



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/SG247148>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG24-7148.

Using the Web material

The additional Web material that accompanies this book includes the following files:

<i>File name</i>	<i>Description</i>
sg247148code.zip	Zipped code samples
corrections7148.txt	Corrections to the book

The previous version of this book, SG24-7148-00, is available in the **Version601** folder.

System requirements for using the Web material

The following system configuration is recommended for our scenario for Modeler, Integration Developer, Process Server, and Monitor. See also “Hardware prerequisites” on page 73.

Hard disk space:	For the base products you require: <ul style="list-style-type: none">▶ Modeler—About 1 GB (product and workspace)▶ Integration Developer—About 6 GB▶ Process Server—About 2 GB▶ Monitor/Dashboard Server —About 5 GB
Operating System:	Windows XP/2000 or Linux®. We only used Windows when writing this book.
Processor:	At least 2 GHz
Memory:	2.0 GB or better

Memory requirements by product

Typical memory requirements for working with our scenario are:

- ▶ Modeler: 400 MB
- ▶ Integration Developer: 500 - 800 MB
- ▶ Process Server: 400 - 500 MB
- ▶ Monitor Server: 700 - 900 MB
- ▶ Portal Server (Monitor Dashboard): 700 - 800 MB

To run Integration Developer together with the Monitor Server test environment, you require a system with at least 2 GB, but more capacity is better.

How to use the Web material

Unzip the contents of the Web material **sg247148code.zip** file onto your hard drive. This creates a directory named **SG247148** with a number of subdirectories.

sampcode	Main directory with sample code
BMPScripts	Monitor Toolkit deployment files
cloudscape	Files for Cloudscape
databases\CLIPTACK	Copy of the initial CLIPTACK database
datasource	JACL script to define the data source
image	Image for generated human task user interface
ldap	Import of groups/users, XSL transformation file
model	Interchange files to import models per chapter
export	Directory for your exports
exportsolution	Export of models and measures
webservice	WSDL file for import into Modeler
monitor	Helper code for Monitor
CrossCellConfiguration	Configuration of SIB Process/Monitor Server
earfiles	Monitor models for deployment
models	Monitor models from Monitor Toolkit
portal	Portal configuration files
security	Custom user registry files
wid	Helper code for Integration Developer Future 1
codesnippets	Helper code
creditRating	Java code for customer credit rating check
dbaccess	Java project with database access code
humanGeneratedUI	Generated humans task user interface tailored
humantaskApp	Custom application for human tasks
humantaskBPC	WAR file with human task custom JSPs
webfront	WAR file with Web front-end application
webservice	WSDL and EAR file with credit check Web service
zF1Interchange	Interchange files of completed applications
widF2	Helper code for WID Future 2
codesnippets	Helper code
humantaskBPC	WAR file with human task custom JSPs
webfrontSCA	WAR file with Web front-end as SCA call
webfrontWS	Web front-end with Web services call
webservice	External Web service to change the interface
zF2Interchange	Interchange files of completed applications
wps	EAR files for installation in Process Server
businessRulesManager	Helper files to install the Business Rules Mgr

The instructions on how to use these files are given in the respective chapters. Here we present only a short extract of major activities.

Creating the CLIPTACK database

The CLIPTACK database holds the customer and order information for the business process.

The CLIPTACK database is automatically created by defining the data source in the server and testing the data source. This can be done under Integration Developer or in the real Process Server:

- ▶ In “Creating a data source for the database” on page 218, we describe how to create the data source manually using the administrative console.
- ▶ In “Using a JACL script to define the data source” on page 404, we describe how to create the data source using a JACL script.

Either method can be used. When the data source is defined, use the test data source facility in the administrative console and the database is created. To populate the database, you have to run the CreateDatabaseServlet as described in “Initializing the database” on page 255.

Importing Modeler solutions

We provide the final business process model for each chapter that deals with the Modeler:

```
SG247148\sampcode\model\Clips And Tacks Current.mar
\Clips And Tacks Current with Simulation.mar
\Clips And Tacks Future 1.mar
\Clips And Tacks Future 1 with Simulation.mar
\Clips And Tacks Future 1 with Measures.mar
\Clips And Tacks Future 2.mar
\Clips And Tacks Future 2 with Simulation.mar
```

These models can be imported into Modeler into separate Modeler projects as described in the respective chapters.

Importing application development solutions

The completed applications of Integration Developer are provided in:

```
SG247148\sampcode\wid\zF1Interchange\ClipsAndTacksF1.zip
\CreditService.zip

SG247148\sampcode\widF2\zF2Interchange\ClipsAndTacksF2.zip
\ClipsAndTacksF2InvokeWS.zip
\CreditServiceExternal.zip
```

These interchange files can be imported into Integration Developer as described in “Importing the human task application” on page 279. Note that any existing code of the same projects is deleted first.

Importing application execution solutions

The completed exported enterprise applications (EAR files) are available in:

```
SG247148\sampcode\wps\ClipsAndTacksF1App.ear          (no security)
  \ClipsAndTacksF1App-LDAP.ear
  \ClipsAndTacksF1App-UserReg.ear
  \ClipsAndTacksF2App.ear                              (no security)
  \ClipsAndTacksF2App-LDAP.ear
  \ClipsAndTacksF2App-UserReg.ear
  \ClipsAndTacksF2InvokeWSEAR.ear
  \ClipsAndTacksFxHumanCustomEAR.ear
  \CreditServiceEAR.ear
  \CreditServiceExternalEAR.ear
```

These EAR files can be installed in a real Process Server for execution after configuring the server with the data source. See Chapter 13, “Deploying and running the application in Process Server” on page 401 for detailed instructions.

Importing application monitoring solutions

The monitor models for development and execution are provided as interchange and EAR files in:

```
SG247148\sampcode\monitor\models\ClipsAndTacksF1BMP.zip
  \ClipsAndTacksF2BMP.zip

SG247148\sampcode\monitor\earfiles\ClipsAndTacksF1BMPEAR.ear
  \ClipsAndTacksF2BMPEAR.ear
  \ClipsAndTacksF2BMPvisualEAR.ear
  generated with visual diagram
```


Abbreviations and acronyms

API	Application program interface	GMT	Greenwich Mean Time
ATM	Automatic teller machine	GUI	Graphical user interface
BAM	Business activity monitoring	HTML	Hypertext Markup Language
BI	Business integration	HTTP	Hypertext Transfer Protocol
BPC	Business process container	IBM	International Business Machines
BPEL	Business Process Execution Language	IDE	Integrated development environment
BPIA	Business process integration and automation	IMS	Information Management System
BPM	Business process management OR business performance management	IP	Internet protocol
CBE	Common business event	IT	Information technology
CEI	Common event infrastructure	ITSO	International Technical Support Organization
CICS	Customer Information Control System	JAAS	Java Authentication and Authorization Service
CPM	Corporate performance management	JACL	Java command language
CRM	Customer relationship management	JDBC	Java Database Connectivity
CVS	Concurrent Versions System	JMS	Java Messaging Service
DAO	Data access object	JNDI	Java Naming and Directory Interface™
DB	Database	JSF	JavaServer Faces
DDL	Data description language	JSP	JavaServer Pages™
DMS	Data movement services	KPI	Key performance indicator
DTO	Data transfer object	LDAP	Lightweight Directory Access Protocol
EAR	Enterprise application archive	LOB	Line of business
EIS	Enterprise information system	LTPA	Lightweight Third Party Authentication
EJB	Enterprise JavaBean	MOM	Message oriented middleware
EPM	Enterprise performance management	MQ	Message queue
ERP	Enterprise resource planning	ND	Network deployment
ESB	Enterprise service bus		
EST	Eastern Standard Time		

OASIS	Organization for the Advancement of Structured Information Standards
OLAP	Online analytical processing
OS	Operating system
PST	Pacific Standard Time
PTF	Program temporary fix
RMI	Remote method interface
SCA	Service Component Architecture
SDO	Service Data Objects
SIB	Service integration bus
SMS	Short message service
SOA	Service-oriented architecture
SOAP	Simple object access protocol
SOX	Sarbanes-Oxley
SQL	Structured query language
SSO	Single sign-on
TKIID	Task instance object identifier
UCM	Universal configuration management
URL	Universal resource locator
USD	US Dollar
UTE	Unit test environment
WAR	Web application archive
WBI	WebSphere Business Integration
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSD	XML schema definition

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbooks publication.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 657. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Building SAO Solutions Using the Rational SDP*, SG24-7356
- ▶ *Web Services Handbook for WebSphere Application Server Version 6.1*, SG24-7257
- ▶ *Rational Business Driven Development for Compliance*, SG24-7244
- ▶ *Best Practices for Using WebSphere Business Modeler and Monitor*, REDP-4159
- ▶ *Getting Started with WebSphere Integration Developer and WebSphere Process Server*, SG24-7130
- ▶ *Patterns: SOA Foundation - Business Process Management Scenario*, SG24-7234
- ▶ *Patterns: Building Serial and Parallel Processes with WebSphere Process Server V6*, SG24-7205
- ▶ *Technical Overview of WebSphere Process Server and WebSphere Integration Developer*, REDP-4041
- ▶ *Build a Business Process Solution using Rational and WebSphere Tools*, SG24-6636
- ▶ *BPEL4WS Business Processes with WebSphere Business Integration: Understanding, Modeling, Migrating*, SG24-6381
- ▶ *Business Integration Management using WebSphere BI Modeler and Monitor A Real World Case Study*, SG24-7024
- ▶ *Business Performance Management . . . Meets Business Intelligence*, SG24-6340
- ▶ *Rational Application Developer V6 Programming Guide*, SG24-6449
- ▶ *Software Configuration Management: A Clear Case for IBM Rational ClearCase and ClearQuest UCM*, SG24-6399

Online resources

These Web sites are also relevant as further information sources:

- ▶ WebSphere Business Integration products Web sites:
 - WebSphere Business Modeler:
<http://www.ibm.com/software/integration/wbimodeler/>
 - WebSphere Integration Developer
<http://www.ibm.com/software/integration/wid/>
 - WebSphere Process Server:
<http://www.ibm.com/software/integration/wps/>
 - WebSphere Business Monitor:
<http://www6.ibm.com/software/integration/wbimonitor/>
- ▶ WebSphere Business Integration Information Center:
<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp>
- ▶ DeveloperWorks:
<http://www.ibm.com/developerworks/websphere>
<http://www.ibm.com/developerworks/websphere/zones/businessintegration/>
- ▶ Business Process Execution Language:
<http://www.ibm.com/developerworks/webservices/library/ws-bpel/>
- ▶ Service Component Architecture and Service Data Objects:
<http://www.ibm.com/developerworks/webservices/library/specification/ws-sca/>
<http://www.ibm.com/developerworks/library/specification/ws-sdo/>
- ▶ Business process management:
<http://www.research.ibm.com/journal/sj/412/leymann.html>
<http://www.bpmi.org/>
http://en.wikipedia.org/wiki/Business_Process_Management
- ▶ Business process integration and automation:
<http://bpia.zurich.ibm.com/>
- ▶ Scalable Vector Graphics (SVG) viewer:
<http://www.w3.org/Graphics/SVG/SVG-Implementations>

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

A

- Action Manager 45, 53
- active instances 35, 480
- Active instances view 465
- activity duration 169
- Adaptive Action Manager 53, 415
 - application 442
 - configuration 441
- administrative console 216
 - Process Server 600
- advanced mode 83
- aggregate metric 191, 202
- alert 35
 - body 444
 - create 362
- Alert Details view 482
- Alerts view 59, 192, 474, 560
 - analysis 481
- Alphablox 55
- analysis
 - dimensional 204
 - simulation 112
- animation 177
- annotation 142, 497
- application
 - adapters 27
 - change 412
 - debugging 299
 - deployment 401
 - development 24
 - solutions 650
 - execution
 - solutions 651
 - export 402
 - implementation 212
 - installation 405, 543
 - monitoring
 - solutions 651
 - programmer 71
 - projects 223
 - run 262, 406
 - startable 419
 - testing 239

- uninstall 412
- assembly diagram 225, 246
 - future 2 511
 - Web service 268
- asynchronous process 138
- attributes
 - technical 504
- authentication
 - mechanism 310
- auto-layout 144
- automatic publishing 216
- automaticApproval 89
- availability matrix 94, 167
- AvailableCredit 90

B

- BAM
 - see Business Activity Monitoring
- benchmark 95
- BOFactory 250
- BPC Explorer 34, 238
 - configuration 259
 - custom JSP 258
 - custom JSPs 265
 - run 406
 - start 241
 - tasks list 242
 - terminate tasks 412
 - work on 262
- BPC Observer 34
- BPEDB database 598
- BPEL
 - process events 411
 - see Business Process Execution Language
- BPM
 - see business process management
- branch percentage 371
- breakpoint 298
 - business process 300
- Breakpoints view 299
- bulk resources 127, 147
- business
 - analyst 70, 78

- comments 142
- flexibility 18
- integration 28
- intelligence analysis 43
- items 78, 80, 88
- object
 - map 536, 539
- objectives 67, 92
- problem 66
- process
 - breakpoint 298
 - export 148
 - integration and automation 149
 - to-be 42
- revision 122
- rule group 230
- rules
 - export 297
- scenario 65
- service
 - objects 21
- services 21
- transformation 4
- Business Activity Monitoring 8, 43
- Business Flow Manager 34
- business innovation and optimization 4
- Business Integration perspective 215, 252, 254, 334
- Business Integration view 224
- business intelligence 14
- business measures 188
 - export 205
 - future 2 506
 - import 329
 - model 40, 188, 190
 - create 194
 - XML file 333
- Business Measures editor 190, 192
- Business Measures view 19, 36, 194
- Business Modeling perspective 82
- Business Monitoring perspective 46, 328
- business object 30, 227
 - map 24, 31
 - relationship 31
- business performance management 7, 13
- business process 4, 16
 - as-is 42
 - component 32
 - container 600
 - model 17
- Business Process Choreographer Explorer 34
- Business Process Choreographer Observer 34
 - database 599
- business process container
 - database 598
- Business Process Execution Language 18, 23, 32, 148, 188
- business process management 3
 - benefits 5
 - definition 4
 - lifecycle 4
 - products 8
 - solution 6
- business rule 24, 33, 150
 - decision table 515
 - future 2 515
 - implementing 229
 - template 231
- business rule group 230
- business rules engine 69
- Business Rules Manager 231, 295
 - JACL script 407
 - using 295, 407
- business state machine 24, 32
- Businessitems.xsd 160
- BusinessSituationName 362, 446

C

- calculateCreditRating 235
- call center 67–68
- Cancel Order and Send Notification task 87
- catch block 541
- CEI
 - see Common Event Infrastructure
- CEI events
 - BPEL 334
 - business object 336
- CEI server 390
 - reboot 439
- change namespace 27, 229
- Check Customer Account Status task 131
- Check Order Handling Policy for Automatic Approval task 131
- CheckCustomerCredit 541
- claim 281
- classification 213, 495
 - dimension 559

- ClipsAndTacks 65
 - current 77
 - Future 1 121
 - human tasks 258
 - improvements 68
 - KPIs 69
 - measurement 477
 - model export 157
 - objectives 67
 - portal page 461
 - problem 66
 - roles 70
 - subsystems 71
- ClipsAndTacksF1
 - project 223
- ClipsAndTacksF1.zip 208
- ClipsAndTacksF1App 223, 402
- ClipsAndTacksF1BMP 338, 490
- ClipsAndTacksF1BMP.mm 339
- ClipsAndTacksF1BPC 258
- ClipsAndTacksF1Database 252
- ClipsAndTacksF1EJB 223
- ClipsAndTacksF1HumanUI 274
- ClipsAndTacksF1Invoke 252, 255, 263
- ClipsAndTacksF1ReviewBMP 373
- ClipsAndTacksF1ShipBMP 373
- ClipsAndTacksF2 510
- ClipsAndTacksF2App 510
- ClipsAndTacksF2BMP 546
- ClipsAndTacksF2BMPvisual.mm 569
- ClipsAndTacksF2EJB 510
- ClipsAndTacksF2Invoke 533
- ClipsAndTacksF2InvokeWS 533
- ClipsAndTacksF2Web 510
- ClipsAndTacksFxHumanCustomEAR 279, 402
- ClipsAndTacksFxHumanCustomWeb 279, 284, 315, 392, 533
- clipstacks.properties 249
- ClipsTacksDAO 248, 251
- CLIPTACK database 212, 650
- Cloudscape 214, 404, 598
- code snippet 520
- collaboration 7
- comment 142
- Common Base Event Browser 408
- Common Business Event 54
- common business events 27, 30
- Common Event Infrastructure 40, 44, 54
- common event infrastructure 28, 30
- comparison analysis 182
- component 10
- composite application development 24
- connection 134
- Console view 216
- context root 528
- continuous process improvement 42
- CopyBO 230
- corporate strategy 92
- cost
 - comparison 184
 - matrix 165
- counter 48
 - create 343
- CreateDatabaseServlet 249, 255, 294
- credit
 - limit 213
 - rating 89, 496
- CreditRating 234, 267, 269
- CreditService 267
- CreditServiceEAR 402
- CreditServiceExternalEAR 541
- CreditServiceWeb 267
- cube 49
- Cube Views 51
- current process 78
- customer
 - classification 489, 495
 - dimension 549
- Customer Information 88
- Customer Record 88
- customer service representative 79, 90
- CUSTOMER table 213
- CustomerDTO 248
- customized JSPs 257
- cycle time 9

D

- dashboard 35, 189
 - alert-instance 563
 - alerts 560
 - colors 469
 - dimensional 471
 - dimensions 556
 - gauge 555
 - gauges 470
 - instances 465, 556
 - KPIs 467, 556

- personalize 468
- reports 561
- views 465
- Dashboard Server
 - installation 622
- data access object 248
- data execution protection 580
- data mart model 49
- Data Mart Model tab 331
- data movement services
 - remove 453
- data pool 240
- data source 215
 - JACL script 404
 - reference 254
 - variable 288
- data transfer object 248
- data types 227
- database
 - business process container 598
 - CLIPTACK 650
 - content 256
 - create 220, 406
 - data source 218
 - DATAMART 45
 - initialize 255
 - MONITOR 45
 - populate 214
 - process choreographer 598
 - replication 484
 - setup 212
- DATAMART database 45, 55, 429
- DATAMART tables 56
- DB2 214
 - AlphaBlox
 - security 642
 - Alphablox 51, 55, 416
 - installation 621
 - command window 388, 423
 - Control Center 56, 427, 478
 - Cube View 416
 - Cube Views 51
 - installation 620
 - replication
 - scripts 449
 - StartApply 450
 - StartCapture 450
 - replication manager 54
 - replicators 416

- db2mdapiclient 426, 453
- debug
 - change variable 300
- Debug perspective 299
- Debug view 299
- Debugger 298
- debugging
 - application 299
- decision 132
 - expression 139
 - probabilities 94
 - table 515
 - template 518
- DeleteServlet 249
- dependencies 255
- Dependencies editor 252
- deployment 401
 - descriptor 316
 - future 2 533
- developerWorks 656
- diagram
 - assembly 225
 - process 226
- Diagram view 60, 192
- dimension 35, 50
- dimensional analysis 204, 471
- Dimensional view 60, 471
 - analysis 483
- Dimensions view 192, 556
- duration
 - comparison 183
 - information 99
 - matrix 93, 166
- dynamic analysis 20, 112

E

- EAR file
 - export 402
- Eclipse 22
- enterprise application
 - installation 418
- enterprise performance management 13
- enterprise service bus 30
- event
 - definition 362
 - inbound 49
 - monitoring 43
 - outbound 49, 365

- Event Model tab 333
- eventCreator 307
- execution failure 541
- exemption period 127
- export
 - business rules 297
 - files 160, 207–208
 - future 2 507
- Export actual values view 474
- Export values view 61
- expression 139
 - builder 140

F

- fact 49, 345
- fault handler 541
- FileRegistrySample 305, 309
- framework 9
- future model 122

G

- gauge 35
- Gauge view 58, 192, 470, 555
 - analysis 481
- global
 - security 308, 320
 - subprocess 502
 - task 499
- groups.props 305

H

- hardware 73
- human task 24, 32, 151
 - API 279
 - application 279
 - run 284, 407
 - security 315
 - application generation 273
 - BPC Explorer 241
 - configuration 237
 - container 603
 - duration 551
 - duration time 100
 - events 372
 - future 2 518
 - implementation 152
 - JSF client 26

- JSPs 259
 - monitor model 372
 - processing logic 260
 - processing time 377
 - security 304, 312
 - testing 314
- Human Task Manager 34, 602

I

- implementation type 150
- import
 - model 83
 - Monitor results 491
 - project interchange file 222
 - WSDL 502
- improvement strategy 495
- inbound event 49
- individual resources 127
- industry standards 17
- information service 26, 287, 520
 - feature 288
 - installation 582
 - test 291
- initialization rule 25
- instance metric 191, 202
- Instances view 57, 556
 - analysis 480
- integrated development environment 23
- Integration Developer
 - see WebSphere Integration Developer
- interface
 - map 24, 31, 536
- interim fixes
 - Integration Developer 582
 - Modeler 577
 - Process Server 596
- InvokeOrderHandling class 250

J

- J2C
 - authentication aliases 639
- JAAS
 - configuration 308
- JACL script 404
- Java
 - snippet 542
- Java activity 233
- Java implementation type 153

- JavaScript 263
- JavaServer Faces 32, 282
- JDBC 292
- JDBC provider 218
- jdbc/cliptack 220, 254, 403
- JMS
 - queue 433
 - queue connection factory 433

K

- keeping processes 227
- key performance indicator 5, 19
- key performance indicators 13, 35, 69, 191
 - ClipsAndTacks 193
 - implement 197
- KPI
 - analysis 495
 - context 50
 - create 353, 357
 - model 50
 - see key performance indicators
- KPI Model tab 331
- KPI view 57, 467, 556
 - analysis 481

L

- launchpad 576
- LDAP 32, 304
 - queries 318
 - query 323, 443
 - security configuration 320
 - server 303, 318, 616
 - staff plug-in provider 318
 - transformation file 322
 - user registry 62, 322
- LDAPTransformation.xml 322
- life cycle 6, 189
- ListServlet 249, 256, 291
- login.jsp 315
- LTPA 310
 - keys 640

M

- map 31
- matrix 92
- measure 50
 - create 344

- tracking key 382
- mediation 31, 33
- memory requirements 648
- merge 134
- metric 48, 191
 - create 347
- Microsoft Visio 83
- mode 83
- model 16
 - export 157
 - future 122, 498
 - import 83, 222
- Modeler
 - feedback loop 379
 - see WebSphere Business Modeler
- modeling
 - mode 82
 - tool 17
- Monitor
 - components 45, 52
 - databases 54
 - export 490
 - Model Manager 53
 - portlets 458–459
 - see WebSphere Business Monitor
 - topology 61
- monitor
 - details model 47
- Monitor Dashboard 45, 57, 458
 - future 2 555
 - Portal Server 429
 - security 636
 - unit test environment 393
- MONITOR database 45, 54
- Monitor Details Model tab 330
- Monitor Development Toolkit 8, 19, 36, 328
 - unit test environment 27
- monitor model 27
 - application 385
 - content 383
 - create 333
 - deployment 384, 416
 - EAR file 384, 415
 - export 396
 - future 2 546
 - generate 338
 - human task 372
 - installation 416
 - layout of XML file 373

- merge 373, 546
- purge 456
- schema remove 454
- stop 451
- terminology 47
- test 392
- testing 552
- undeployment 396, 451
- uninstall 455
- version 569
 - deployment 570
- visual 564
- Monitor Model Editor 27, 40, 46, 339
- Monitor Server 52
 - installation 622
 - security 636, 639
 - service integration bus 632
 - unit test environment 384
- MonitorAlphabloxAlias 385
- monitored values 196
- Monitoring
 - Programming Model 46
- monitoring context 48, 340
- monitoring events
 - generate 337
- Monitoring Flow view 343
- multidimensional analysis 35

N

- namespace
 - business items 229
 - business objects 27
 - change 513
- notification 53
 - template 443

O

- OASIS 30
- OLAP Center 427
- Online Application timetable 125
- operation type 156
- Order 67, 88
- Order Counter 202
- Order Handling (Current) 86
- Order Handling (Future 1) 124, 129
- Order Handling (Future 1) Business Measures.m 207
- Order Handling (Future 2) 496, 500

- measurement 545
- order handling process 68
 - current 77
- Order Item 88
- order manager 68, 71, 90
- Order Processing Data 88
- OrderDTO 248
- OrderHandlingFuture1.bpel 228
- ORDERITEM table 214
- OrderItemDTO 248
- ORDERMAIN table 214
- OrderStatus 89
- organization 36
- Organization view 58, 192
- otherwise clause 25
- outbound event 49, 365
- output logic probability 103
- Overall Order (Future 1) 145

P

- perspective 82
- Physical Resources view 228
- port number 534, 587
- portal
 - administrator 459
 - authorization 475
 - page 461
- portlet 458
 - wiring 562
- Potential Instance Creator 312
- Potential Owner 238, 312
- prerequisites 73
- probability 94
 - information 102
- probability matrix 167
- Problems view 224
- process 4
 - analysis 91
 - assessment 92, 164
 - case 112, 114
 - analysis 115
 - summary 179
 - cases summary 114
 - comparison analysis 182
 - cost analysis 118, 181
 - diagram 36, 226
 - future 2 511
 - duration

- analysis 112, 178
- flow 78–79
- global 499
- invocation 250
- modeling
 - enhancements 19
- see also business process
- simulation 91–92
- test facility 239
- testing 239
- validation 143
- Process Server
 - configure CEI 433
 - LDAP 318
 - security 306, 404
 - see WebSphere Process Server
 - test environment 215
- Product Shipment 89
- PRODUCT table 213
- ProductDTO 248
- project
 - interchange file 158, 222
- project interchange file 222
- Project Tree view 84
- Properties view 225, 227
- purge 456

R

- rating 89, 213
- Rational Application Developer 23
- Rational Product Updater 26, 581
- Rational Software Architect 23
- Receive Order process 87, 122
- Redbooks Web site 657
 - Contact us xxii
- reference 30
 - architecture 10
- relationship 24, 31
- Reports view 59, 192, 561
- repository tables 54
- requirements 17
- RequisitePro 25
- resource
 - availability 99
 - durations 170
 - pool 505
 - usage 116, 181
- Resource perspective 82

- resources 81, 92, 138
 - bulk 127
 - individual 127
 - role 97
- revenue 9
- Review Order task 87
- reviewOrderInput.jsp 261
- reviewOrderOutput.jsp 261
- risk 9
- RMI 216
- roles 70
 - duration 100
 - matrix 92, 165
- rolling time period 358
- round-trip measures 369
 - human tasks 372
- rule group 150
 - implementation 151

S

- sampleldapconfiguration 324
- Sarbanes-Oxley 14
- SCA
 - see Service Component Architecture
- scorecards 13
- SDO
 - see Service Data Objects
- security
 - certificate 314
 - constraint 316
 - groups and users 304
 - human task 304
 - Integration Developer 311
 - Process Server 603
 - remove 325
 - roles 306, 608
 - user registry 308
- selector 24, 31
- server
 - add project 239
 - configuration 217
 - log 408
- Servers view 216, 239, 311
- service 29
 - component 29, 32
- Service Component Architecture 3, 11, 23, 29, 250
 - component 29
 - implementation 30

- interface 29
 - module 33
 - specification 13
- Service Data Objects 12, 250
 - business objects 30
 - specification 13
- service integration bus 54, 632
 - post-configuration 635
- ServiceManager 250
- service-oriented architecture 9
 - benefits 9
 - foundation 10
- services 24
- Services Data Objects 3
- sg247148code.zip 648–649
- Ship Order to Customer task 87
- shipOrderInput.jsp 262
- shipOrderOutput.jsp 262
- shipper 90
- simulation 18
 - analysis 112
 - attributes 96, 108, 168, 176
 - compare Monitor results 495
 - control panel 109, 177
 - current 91
 - defaults 104
 - environment 96, 168
 - future 1 163–164
 - future 2 505
 - information 103
 - preferences 96
 - profile 104–106
 - results
 - current 111
 - future 177
 - future 2 505
 - run
 - current 109
 - future 177
 - settings 110
 - snapshot 92
 - current 103
 - future 172
 - token 95
- situation event 446
 - binding 446
- six sigma 14
- snapshot 92
- snippet 520, 542
 - SQL 289
 - visual 522
- SOA
 - see service-oriented architecture
- SOAP 216
- soap
 - address location 535
- SQL
 - snippet 289
- staff 17
- staff plug-in 313
 - LDAP 318
 - security 519
- stand-alone reference 245
- standards 17
- Start node 147
- state machine 154
 - implementation 155
- state tables 54
- static analysis 101
- Step 1. Run Data Services Generation 387, 420
- Step 2. Run Schema Create Scripts 388, 422
- Step 3. Run DMS Create Scripts 424
- Step 4. Import DB2 Cube Definition 426
- Step 5. Manage Alphablox Cube 429
- Step 6. Configure CEI Distribution 389, 431
- Step 7. Confirm CEI Server Reboot 390, 439
- Stop node 147
- stopwatch 48
 - create 353
- strategy 495
- subsystems 72
- SVG diagram 564
- svg file 207
- system administrator 71
- system architect 71

T

- table 213
- task 131
 - duration 20
 - local 499
- TaskQuery 280
- technical attributes 149, 504
- Technical Attributes view 151
- technology adapters 27
- template rule 231
- test

- connection 221
- environment 215
- testing 239
- timetable 81, 125
- Tivoli Directory Server 318
 - installation 612
- token 95
- tracking key 379, 490
- transform
 - business process 17–18
- trigger 48
 - create 341
 - source 342
- troubleshooting 456
- tutorial 83

U

- unit test environment 36, 51, 328
 - dashboard 393
- user registry 32, 304
 - files 306
- USER_INSTALL_ROOT 220
- users 304
- users.props 305

V

- valid from date 227
- validation 101, 143
- Variables view 299
- Visio 83
- visual
 - process instance model 568
 - snippet 25, 522, 542
- Visual KPI model 567
- visual model 564
 - instance metrics 564
 - KPIs 566
 - path coloring 565
- Visual Model tab 332

W

- WBM Web Dashboard 393, 553
- Web application
 - architecture 247
 - import 252
 - invokes process 254
 - run 262

- Web front-end application 247
- Web perspective 215, 252
- Web service 30, 267
 - binding 534
 - deployment 270, 402
 - import 267, 502, 536
 - invocation 269
 - testing 270
- Web Services Description Language 23
- WebSphere Application Server 28
 - Network Deployment 28
- WebSphere Business Integration Server Foundation 82, 152
- WebSphere Business Modeler 8
 - import Monitor data 491
 - installation 576
 - Publishing Server 17
 - Version 6.0.2 21
 - Version 6 16
 - Version 6.0.2 19
- WebSphere Business Monitor 8
 - architecture 44
 - dashboard views 192
 - functionality 40
 - installation 619
 - launchpad 621
 - prerequisites 620
 - Version 6.0.2 36, 40
- WebSphere Business Monitor Development Toolkit 27
 - installation 584
- WebSphere ESB 30
- WebSphere Integration Developer 8
 - implement future 2 509
 - import 222
 - installation 578
 - Version 6 22
 - Version 6.0.2 25
- WebSphere MQ Workflow 82
- WebSphere Partner Gateway 33
- WebSphere Portal Server 35
 - security 636
- WebSphere process integration 15
- WebSphere Process Server 8, 82
 - administrative console 600
 - configuration 403
 - installation 588
 - mod 83
 - mode 148

- service integration bus 634
 - Version 6 28
 - Version 6.0.2 33
- WebSphere Service Registry and Repository 27
- wikipedia 5
- wiring views 61
- work item 26
- wsadmin command 439
- WSDL
 - import 502

X

XML

- data source 12
- schema definition 23

XSL

- transform file 319



Redbooks

Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products

(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages



Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products



Model and simulate with Business Modeler

This IBM Redbooks publication presents a business process management (BPM) “improvement cycle” scenario, showing how a business can use a full business integration solution to complete the following tasks:

Implement with Integration Developer and Process Server

- ▶ Model and simulate a business process
- ▶ Develop and test an application to implement the business process
- ▶ Deploy and run the application on a server
- ▶ Implement and test business measures
- ▶ Monitor the application to observe pre-determined key performance indicators
- ▶ Import the observed data to make revisions to the original process model

Measure with Business Monitor

The business scenario described in this document has been simplified to provide a full description of each stage of the BPM end-to-end process. To avoid having an overly large and unwieldy document, the authors focus on specific tasks, elements, and details, and not on presenting all possible facets of a complex business process.

This document updates the IBM Redbooks publication SG24-7148-00 using the Version 6.0.2 products of WebSphere Business Integration.

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