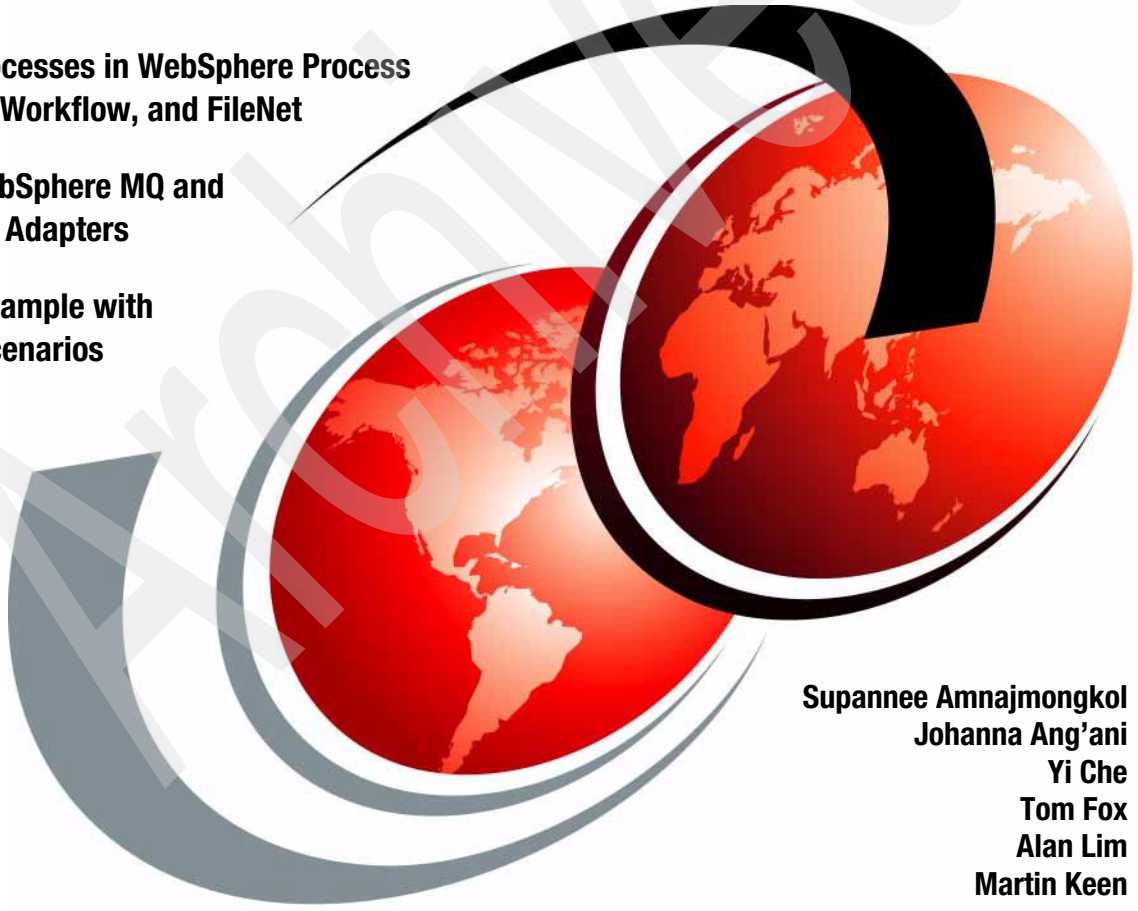
WebSphere® software

IBM

Business Activity Monitoring with WebSphere Business Monitor V6.1

- Monitor processes in WebSphere Process Server, MQ Workflow, and FileNet
- Monitor WebSphere MQ and WebSphere Adapters
- Learn by example with practical scenarios



Supannee Amnajmongkol
Johanna Ang'ani
Yi Che
Tom Fox
Alan Lim
Martin Keen

Archived

Redbooks

ibm.com/redbooks

Archived

International Technical Support Organization

Business Activity Monitoring with WebSphere Business Monitor V6.1

July 2008

Archived

SG24-7638-00

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

Archived

First Edition (July 2008)

This edition applies to IBM WebSphere Business Monitor V6.1.

© Copyright International Business Machines Corporation 2008. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices xi
Trademarks xii

Preface xiii
The team that wrote this book xiv
Become a published author xvi
Comments welcome xvii

Part 1. Overview of WebSphere Business Monitor 1

Chapter 1. Business activity monitoring, the Common Event Infrastructure, and Common Base Events 3
1.1 Business activity monitoring 4
1.1.1 Business activity monitoring components 6
1.2 Common Event Infrastructure 6
1.2.1 Common Event Infrastructure components 7
1.3 Common Base Event model 8

Chapter 2. WebSphere Business Monitor and concepts 11
2.1 Introduction to WebSphere Business Monitor 12
2.2 WebSphere Business Monitor V6.1 architecture 13
2.2.1 WebSphere Business Monitor V6.1 buildtime architecture 14
2.2.2 WebSphere Business Monitor V6.1 runtime architecture 24
2.3 New in WebSphere Business Monitor V6.1 34
2.3.1 Development toolkit 34
2.3.2 Monitor server 39
2.3.3 Dashboards 44
2.3.4 Installation and administration 52
2.4 Summary 54

Chapter 3. Using WebSphere Business Monitor 55
3.1 Installation deployment topology for WebSphere Business Monitor 56
3.1.1 Development topology 56
3.1.2 Single server topology 57
3.1.3 Distributed topology 59
3.2 Installing the toolkit 62
3.2.1 System requirements 64
3.2.2 Installing the toolkit into WebSphere Integration Developer 64
3.2.3 Adding and removing the Monitor test environment 67

© Copyright IBM Corp. 2008. All rights reserved.

3.2.4 Verifying the toolkit installation 68
3.3 Installing WebSphere Business Monitor 70
3.3.1 WebSphere Business Monitor prerequisites 71
3.3.2 Preparing to install the required software 71
3.3.3 Installing WebSphere Business Monitor on a single server 78
3.3.4 Verifying the installation 88
3.3.5 Starting and stopping the WebSphere Business Monitor server 89
3.4 Deploying and administering monitor models 90
3.4.1 Installing a monitor model by using the default settings 91
3.4.2 Installing a monitor model by using the remote CEI 97
3.4.3 Starting and stopping a monitor model 103
3.4.4 Versioning a monitor model 104
3.4.5 Updating a monitor model 108
3.4.6 Removing a monitor model and data 109
3.5 Monitoring with Web-based dashboards 115
3.5.1 Modes in your dashboard 116
3.5.2 Creating a dashboard 117
3.5.3 WebSphere Business Monitor KPI manager 128
3.6 Summary 136

Part 2. Business activity monitoring 137

Chapter 4. Business scenario: ClipsAndTacks 139
4.1 Case study: ClipsAndTacks Office Supplies Pte. Ltd. 140
4.1.1 Background 140
4.1.2 Business challenges 140
4.2 Order handling process 143
4.3 Contract management process 145
4.4 Order database 146
4.5 Shipping 147
4.6 Credit check 148
4.7 Summary 148

Chapter 5. WebSphere Process Server 149
5.1 Introduction to WebSphere Process Server 150
5.1.1 SOA core 150
5.1.2 Supporting services 151
5.1.3 Service components 151
5.2 Overview of the integration of WebSphere Process Server into WebSphere Business Monitor 152
5.3 Integrating WebSphere Process Server into WebSphere Business Monitor 153
5.3.1 Software requirements 154
5.3.2 Importing the ClipsAndTacks BPEL process 154

5.3.3 Exploring the business process 158
5.3.4 Defining CEI events for the BPEL process 162
5.3.5 Developing the monitor model by using the WebSphere Business Monitor Toolkit 169
5.3.6 Testing the integration. 216
5.3.7 Configuring the action manager for business situation events 225
5.4 Summary 247

Chapter 6. WebSphere ESB and IBM Adapters 249
6.1 Business activity monitoring with IBM Adapters 250
6.1.1 IBM Adapters 250
6.1.2 WebSphere ESB 251
6.1.3 IBM Adapters and WebSphere ESB 252
6.1.4 Service component monitoring 252
6.1.5 Event generation options 254
6.2 Scenario description 255
6.3 Preparing the environment for the sample scenario 256
6.3.1 Setting up the development environment 256
6.3.2 Setting up the EIS 256
6.4 Scenario 1: Monitoring the built-in service component events 258
6.4.1 Creating a module that contains the WebSphere Adapter 260
6.4.2 Configuring the server resources 277
6.4.3 Testing the adapter 289
6.4.4 Enabling event generation 291
6.4.5 Validation 293
6.4.6 Creating the monitor model. 295
6.4.7 Creating the dashboard. 308
6.4.8 Testing the service component monitoring scenario. 310
6.5 Scenario 2: Monitoring mediation flow custom events 311
6.5.1 Creating the mediation module 312
6.5.2 Enabling event generation 327
6.5.3 Testing the Event Emitter primitive scenario. 341
6.6 Adding the Percentage of Large Orders KPI. 342
6.6.1 Cleaning up the dashboard. 346
6.6.2 Troubleshooting: NameNotFoundException error messages 346
6.7 Design considerations 347
6.7.1 When to monitor events from the event sources directly 347
6.7.2 When to use built-in service component events and Event Emitter primitive 348
6.7.3 Where to use the Event Emitter primitive in the mediation flow . . . 348

Archived

6.8 Summary 350

Chapter 7. FileNet P8 BPM 351
7.1 Introduction to FileNet P8 BPM 352
7.2 Overview of integrating FileNet P8 BPM with WebSphere Business Monitor 354
7.3 Runtime topology for the sample 355
7.4 Deploying the FileNet P8 BPM Common Base Event Adapter to the WebSphere Business Monitor server 357
7.4.1 Creating the BPMSTATE database 358
7.4.2 Creating the work manager 359
7.4.3 Creating the database authentication aliases 362
7.4.4 Creating the JDBC data sources 366
7.4.5 Installing the FileNet P8 BPM Common Base Event Adapter 375
7.4.6 Setting optional configuration properties 377
7.4.7 Starting the FileNet P8 BPM Common Base Event Adapter for WebSphere 380
7.5 FileNet P8 BPM Common Base Event events 381
7.6 FileNet P8 BPM Monitor Model for WebSphere Business Monitor 384
7.6.1 Monitor model 384
7.6.2 Process monitor context 384
7.6.3 Work item monitor context 386
7.6.4 Data mart model 387
7.7 Implementing the FileNet P8 BPM processes 387
7.7.1 Implementing the Contract Management Workflow 388
7.8 Running the Contract Management Workflow 404
7.9 Creating the FileNet P8 BPM Monitor Model for WebSphere Business Monitor 417
7.9.1 ClipsAndTacks KPIs and measures 417
7.9.2 Importing the monitor model 417
7.9.3 Updating the event definition 424
7.9.4 Creating metrics 425
7.9.5 Creating measures 428
7.9.6 Creating dimensions 430
7.9.7 Creating inbound events 433
7.9.8 Creating stopwatches 436
7.9.9 Creating KPIs 439
7.9.10 Defining the visual model 445
7.9.11 Defining actions 447
7.9.12 Creating situation events 450
7.10 Generating and deploying the monitor model 453
7.10.1 Generating the Monitor J2EE projects 454
7.10.2 Generating the EAR file 456

7.10.3 Deploying the monitor model 458
7.11 Configuring action manager for business situation events 458
7.12 Testing the monitor model 462
7.13 Creating a dashboard 462
7.14 Best practices for monitoring FileNet P8 BPM using IBM WebSphere Business Monitor 478
7.14.1 FileNet P8 BPM Monitor Model for WebSphere 478
7.14.2 Process engine software 479
7.14.3 DB2 database for Monitor server 479
7.14.4 FileNet P8 BPM Monitoring Dashboard for WebSphere Business Monitor 479
7.15 Summary 479

Chapter 8. WebSphere MQ Workflow 481
8.1 Introduction to WebSphere MQ 482
8.2 Overview of WebSphere MQ Workflow support in WebSphere Business Monitor V6.1 483
8.2.1 Architecture of WebSphere MQ Workflow support 483
8.2.2 Installation topologies 486
8.2.3 Prerequisites for WebSphere MQ Workflow support 488
8.3 Setting up the runtime environment 489
8.3.1 Runtime topology for the sample 489
8.3.2 Applying WebSphere MQ Workflow ServicePack 6 490
8.3.3 Applying WebSphere MQ Workflow SupportPac WA61 494
8.4 Implementing the scenario by using WebSphere MQ Workflow 503
8.4.1 Implementation of the WebSphere MQ Workflow process 504
8.4.2 Use cases of the process 506
8.4.3 Setting the queue and queue manager name 507
8.4.4 Enabling the audit to WebSphere MQ 508
8.5 Developing the monitor model 509
8.5.1 Setting up the monitor model development environment 509
8.5.2 Developing the monitor model 519
8.6 Deployment and monitoring 574
8.6.1 Deploying the WebSphere MQ Workflow process 575
8.6.2 Deploying the monitor model 576
8.6.3 Exercising the WebSphere MQ Workflow process 581
8.6.4 Configuring the dashboard views 586
8.7 Summary 597

Chapter 9. Custom event emitter 599
9.1 Monitoring arbitrary events 600
9.1.1 Approaches for emitting Common Base Events 600

9.2 Event emitter concepts 601
9.2.1 Using the emitter framework 601
9.2.2 Retrieving EventFormatter from the EmitterFormatterFactory 603
9.2.3 Invoking the EventFormatter or creating Common Base Events . . 603
9.2.4 Retrieving an emitter from an EmitterFactory 605
9.2.5 Emitting the Common Base Event to the CEI server 605
9.3 Scenario description 606
9.4 Design overview of the emitter application 607
9.5 Importing and working with the source code project 609
9.5.1 Importing the project 610
9.5.2 Creating the shipment notification event formatter 616
9.6 Configuring and deploying the emitter application 627
9.6.1 Setting up the WebSphere Business Monitor server scheduler . . . 627
9.6.2 Deploying the sample event emitter application 640
9.6.3 Testing the emitter 641
9.7 Creating the monitor model 642
9.7.1 Creating the dashboard 658
9.7.2 Testing the custom event emitter sample 659
9.8 Design considerations 660
9.8.1 Custom event emitters and WebSphere Adapters 661
9.8.2 Creating stand-alone emitter applications or modifying existing applications 661
9.8.3 Choosing Common Base Events and XSD definitions 662
9.9 Summary 663

Chapter 10. WebSphere MQ 665
10.1 Introduction to WebSphere MQ 666
10.2 Overview of integrating WebSphere MQ with WebSphere Business Monitor 667
10.3 The business challenge 669
10.3.1 Outside the managed process 669
10.3.2 Integration topology 671
10.4 Building the scenario 672
10.4.1 Setting up WebSphere MQ 672
10.4.2 Establishing the MQ to CEI bridge 678
10.4.3 Creating a monitor project 686
10.4.4 Defining the events 689
10.4.5 Defining the monitor model 693
10.4.6 Generating the deployment artifacts 719
10.4.7 Deploying the monitor application 721
10.4.8 Defining the business dashboards 725
10.4.9 Testing by using ad hoc event generation 729

10.5 Summary 734

Part 3. Appendixes 735

Appendix A. Security activation for monitoring WebSphere Process Server human tasks 737
Securing WebSphere Process Server and WebSphere Business Monitor . . 738
Enabling event emitting for WebSphere Process Server human tasks. . . 740
Installing the Global Human Task monitor model 741
Connecting Business Process Choreographer with a WebSphere Business Monitor dashboard 744
Enabling security for WebSphere Business Monitor 747
Enabling security for WebSphere Business Monitor applications 760
Updating the Alphablox Server.properties file 764
Updating the J2C authentication data entries for messaging buses for WebSphere Business Monitor 764
Enabling security for WebSphere Business Monitor resources 769
Modifying the Alert User IDs 770
Enabling data security for monitor models 771
Enabling security for WebSphere Process Server 773
Enabling security for WebSphere Process Server applications 782
Updating the J2C authentication data entries for messaging buses for WebSphere Process Server 786
Granting access to the foreign queues of monitor models 790
Configuring server-to-server SSL from WebSphere Business Monitor to WebSphere Process Server 792
Configuring the dashboard for human task monitoring 795
Creating a Human Tasks view 796
Advanced features of the Human Tasks view 797

Archive

Summary 800

Appendix B. Additional material 801
Locating the Web material 801
Using the Web material 802
How to use the Web material 802

Abbreviations and acronyms 803

Related publications 805
IBM Redbooks 805
Online resources 805
How to get Redbooks 806
Help from IBM 806

Index 807

Archived

Notices

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

© Copyright IBM Corp. 2008. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

AIX 5L™	DB2®	Redbooks (logo) ®
AIX®	developerWorks®	SupportPac™
AlphaBlox®	FileNet®	Tivoli®
CICS®	IBM®	WebSphere®
Cube Views™	Informix®	z/OS®
DataPower®	Rational®	
DB2 Universal Database™	Redbooks®	

The following terms are trademarks of other companies:

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

FileNet, and the FileNet logo are registered trademarks of FileNet Corporation in the United States, other countries or both.

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

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

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

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

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

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

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.

Archived

Preface

Business activity monitoring (BAM) is a solution that provides real-time tracking of business events, including the tracking of business processes, operational activities, and business situations. The business events that drive business activity monitoring can be sent by a variety of applications and technologies. The collected event information is available in the form of dashboards.

Through step-by-step hands-on examples, in this IBM® Redbooks® publication, we explore how you can use IBM WebSphere® Business Monitor V6.1 to track business events from a variety of applications. We discuss combining WebSphere Business Monitor with process-based applications running in IBM WebSphere Process Server, WebSphere MQ Workflow, and FileNet® P8 BPM to achieve business activity monitoring.

In addition, we explore monitoring business events from non-process based applications including WebSphere MQ, and WebSphere Adapters running within WebSphere Enterprise Service Bus (ESB). We also explore how to create a custom event emitter.

The motivation in writing this book was to show how WebSphere Business Monitor can be combined with a wide variety of other IBM products to achieve BAM. Whether you are new to BAM and WebSphere Business Monitor, or you already have experience with them, you will find this book helpful.

To help demonstrate BAM, we provide detailed step-by-step examples for numerous products. If you are looking to add BAM to an existing business process, then the chapters on WebSphere Process Server, WebSphere MQ Workflow, and FileNet P8 BPM should be of interest to you. In this book we also look at nonprocess-based applications, with chapters on WebSphere MQ, WebSphere Adapters with WebSphere ESB, and a custom event emitter written in Java™.

Throughout this book, we use a common business scenario to illustrate BAM. This business scenario is based on a fictional company called *ClipsAndTacks Office Supplies Pte. Ltd*. We hope you can identify with some of the business challenges and solutions described in this book for ClipsAndTacks and that you can apply them to your own business.

We hope that this book provides you with a starting point for building your own BAM solutions.

© Copyright IBM Corp. 2008. All rights reserved.

The team that wrote this book

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

Supannee Amnajmongkol is a WebSphere Technical Sales Leader for IBM Software Group in Thailand. She is responsible for all major WebSphere service-oriented architecture (SOA), Business Integration, and Business Process Management opportunities in Thailand. She achieved IBM ASEAN/SA IT Specialist Advisory Accreditation in 2006. She is an IBM Certified SOA Solution Designer and a Certified Solution Developer and Designer for WebSphere Integration Developer and WebSphere Business Monitor. Supannee holds a master degree in information technology from Swinburne University of Technology in Australia.

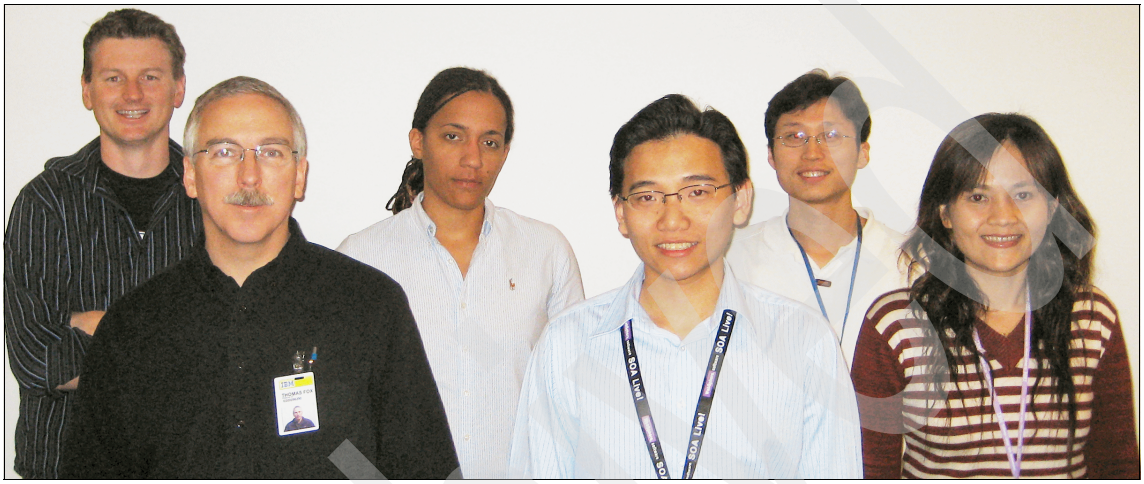
Johanna Ang'ani is an IT Specialist working for the WebSphere Solution Center in the development laboratory in Boeblingen, Germany. Johanna provides expert knowledge about the newest versions of WebSphere Process Server and works with customers in that area. She holds a diploma degree in computer science from the University of Applied Science in Berlin (TFH Berlin). Before joining the WebSphere Solution Center, she worked as a software developer for IBM Research (T.J. Watson Research laboratory) and IBM Deutschland Entwicklung GmbH (Boeblingen, Germany).

Yi Che is a software developer at the IBM China Development Lab in Beijing, China. He has two years of experience in the WebSphere Business Monitor field and now is responsible for the development of the new features of WebSphere Business Monitor. Prior to this, Yi Che worked on WebSphere Business Integration Adapter for two years, providing L3 support. Yi Che holds a master degree in computer science from the Tsinghua University in China.

Tom Fox is a technology advisor at Wachovia Bank in Charlotte, NC. He has over 25 years of experience in systems development and various other IT fields. His areas of expertise include messaging and integration technologies. Tom holds a bachelor degree in computer science from the Georgia Institute of Technology.

Alan Lim is a WebSphere IT Specialist with the IBM Software Group in Singapore. He advises customers and business partners on the design and implementation of SOA and business process management (BPM) solutions. He is also responsible for articulating the technical benefits of IBM middleware. Alan is an IBM Certified SOA Solution Designer and a Certified Solution Developer and Designer for WebSphere Integration Developer and WebSphere MQ. He holds a bachelor degree in electrical engineering from the National University of Singapore.

Martin Keen is a Senior IT Specialist at the ITSO, Raleigh Center. He writes extensively about WebSphere products and SOA. He also teaches IBM classes worldwide about WebSphere, SOA, and ESB. Before joining the ITSO, Martin worked in the EMEA WebSphere Lab Services team in Hursley, UK. Martin holds a bachelor degree in computer studies from Southampton Institute of Higher Education.



The team from left to right: Martin, Tom, Johanna, Alan, Yi Che, and Supannee

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

John Alcorn
BAM Platform lead architect

Victor Chan
BAM Dashboard and Web Interface development lead architect

Phuah Mong Jen
Senior System Consultant ECM, IM, ASEAN

Wei Liao
Software engineer for IBM FileNet BPF development in IBM CSDL

Jane Luo
BPM Developer, Software Developer: 3GL Programming Languages, IBM Software Group, Information Management

Jim McGarrahan
WebSphere Business Process Management Development

Archive

Steve Myers
WebSphere Business Monitor

Martin Phillips
WebSphere MQ

Latha Sivakumar
WebSphere Business Monitor Development

Scott Walden
AIM Early Programs - WebSphere BPM and Connectivity

Ning Wang
Staff software engineer in IBM China Software Development Lab

Eric Wayne
Business Activity Monitoring lead architect

Lai Feng Zhang
FileNet BPF developer in IBM China Software Development Lab

Become a published author

Join us for a two- to six-week residency program! Help write a book 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

Archived

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM 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

Archived

Archived

Part 1

Overview of WebSphere Business Monitor

In this part, we provide an introduction to the concepts and procedures that you need to perform BAM. It contains the following chapters:

- Chapter 1, “Business activity monitoring, the Common Event Infrastructure, and Common Base Events” on page 3

  This chapter provides a brief overview of BAM, and introduces the important concepts of CBE and CEI.

- Chapter 2, “WebSphere Business Monitor and concepts” on page 11

  This chapter describes the features of WebSphere Business Monitor and includes a section on the new features added to WebSphere Business Monitor V6.1.

© Copyright IBM Corp. 2008. All rights reserved.

- Chapter 3, “Using WebSphere Business Monitor” on page 55

  This chapter provides step-by-step instructions for installing WebSphere Business Monitor and the related toolkit. It also explains how to work with monitor models and dashboards.

Archived

1

Business activity monitoring, the Common Event Infrastructure, and Common Base Events

In this chapter, we provide an overview of business activity monitoring (BAM). We describe the general notion and components of BAM. We also provide a brief introduction to the Common Base Event and the IBM Common Event Infrastructure (CEI). A general understanding of both Common Base Event and CEI will help you in understanding the examples in this book.

© Copyright IBM Corp. 2008. All rights reserved.

1.1 Business activity monitoring

The term *business activity monitoring* was introduced by Gartner, Inc. as their notion of aggregation, analysis, and real-time presentation functionality of business relevant information.

Today's companies face fast growing IT infrastructures, which are technology heterogeneous and highly distributed. Business activities running within these landscapes are more automated and contain an increasing size of business relevant data.

To keep control of their business and to quickly react to new strategy targets, business problems, and market risks and opportunities, it is crucial for them to have a real-time overview of all relevant business activities. The lack of such a real-time information system can increase the time lapse between the occurrence and the notification of business-critical events resulting in long and expensive reaction times. A well-defined processing of business events, best based on a standardized underlying technology, enables companies to improve the pace and effectiveness of their business operations.

BAM is a solution that provides this real-time view to data collections of business events by aggregating and presenting real-time, role-based information, including tracking of business processes, operational activity and business situations. It is driven by business events, which can be sent by a variety of applications and technologies. The collected event information is available in the form of *dashboards*. Dashboards show operational and financial data, for example metrics or key performance indicators (KPIs), and compare the current state with defined target goals. As a result, companies can address problem areas quickly and efficiently.

The purpose of BAM is to monitor occurring business events and provide the basis to adjust and improve existing business processes. The measurement and analysis of business processes are fundamental to continuous process optimization and cost-cutting. An example for such a process optimization is the adjusting of staff workload.

BAM is relevant in all areas of business. A financial institution, for example, may use it to manage and track loan processes in real-time, combining information about human and automated elements of the process into a single view. A government can leverage BAM to gain visibility into the operations of a social services agency department. In healthcare, BAM can be used to gain an overview of all operations within a hospital, including the management of insurance claims processing, scheduling of testing equipment needs, and staff assignments.

An important aspect of BAM is the role-based display of business events. Effective and quick actions can only be taken if the monitoring results are tailored to the need and knowledge of the current user. There is a distinction between the following types of roles:

- Business leader and executives

  Persons in this role need an insight into operational performance in order to make good business decisions. They need an overview about KPIs and are interested in reports. A drill down of a general set of information to specific areas of interest is important for them.

- Business process management (BPM) business experts

  BPM business experts are business architects and analysts who are responsible for the correct succession of business processes. They need an overview of the operational performance in order to improve and adjust processes. They are interested in KPIs, alerts, and reports, and need the ability to drill down information for deeper analysis.

- Process users

  Process users use BAM as a working environment. They receive alerts, can forward them to other users, work on human tasks, and are interested in an overview of running instances.

- IT experts

  Even though BAM is focused on business events, it is also of interest for IT experts. They must be aware of the monitored data so that they can adapt IT environments to fulfill the IT part for the business requirements.

Because BAM often leads to confusion, it is important to distinguish it from technical monitoring. BAM is focused on events emitted in a business context. Most of the time, the important information is in the payload of an event.

Technical monitoring focuses on events and exceptions emitted by IT resources. In many cases, the pure events are enough. Technical monitoring provides information about the availability and performance of IT resources to check that systems and applications run as designed. In order to follow its intention, technical monitoring requires a high number of events. The importance for BAM is often not the quantity of incoming events but the quality of the event payload. The target audience for technical information provided by technical monitoring is IT roles such as support, maintenance, and IT administration staff.

The borders between these two kinds of monitoring are smooth, but there are overlaps. They may even be combined to provide an insight into the performance and improvement needs of a business as a whole.

Archive

The distinction between technical monitoring and BAM is important for the selection of the appropriate product. IBM offers a BAM solution called *WebSphere Business Monitor*, which we discuss in this book. WebSphere Business Monitor is capable of monitoring events generated by all kinds of technologies that can produce CEI events. Technical monitoring capabilities are provided by several IBM Tivoli® products, such as Tivoli Composite Application Manager.

1.1.1 Business activity monitoring components

BAM consists of the following components independent from a specific BAM solution:

- Business event

  A business event indicates the occurrence of an event that is important to an operation of a business activity. It is different from an IT event, which shows the occurrence of an event in the context of an IT resource.

- Business activity

  A business activity provides services or functionality that is required to fulfill a specific business need.

- Metric

  A metric is a measurement of a specific property element of a business activity that can be used to monitor business operations.

- Key performance indicator

  KPIs are metrics that are used to measure the progress of an enterprise goal. KPIs have to be measurable as well as achievable and provide the possibility to aggregate measures over a set of instances. In order to define KPIs, companies must have clear performance requirements, and they must also have a defined strategic target that can be compared to runtime values.

1.2 Common Event Infrastructure

CEI is an embeddable technology that is intended to provide basic event management services to applications that require those services. This event infrastructure serves as an integration point for consolidation and persistence of raw events from multiple, heterogeneous sources and distribution of those events to event consumers. Events are represented by using the Common Base Event model, which is a standard, XML-based format that defines the structure of an event.

Archived

By using this common infrastructure, diverse products that are not tightly coupled with one another can integrate their management of events, providing an end-to-end view of enterprise resources and correlating events across domain boundaries. For example, events that are generated by a network monitoring application can be correlated with events that are generated by a security application. Such correlation can be difficult to achieve when each product uses its own approach to event management.

CEI provides facilities for generation, propagation, persistence, and consumption of events. However, it does not define the events themselves. Instead, application developers and administrators define event types, event groups, filtering, and correlation.

1.2.1 Common Event Infrastructure components

The CEI consists of the several major components.

Common Base Event

The Common Base Event component supports the creation of events and access to their property data. Event sources use the Common Base Event APIs to create new events conforming to the Common Base Event model. Event consumers use the APIs to read property data from received events. In addition, applications can convert events to and from XML text format, supporting interchange with other tools. The Common Base Event component is part of the Eclipse Test and Performance Tools Platform (TPTP).

Emitter

The emitter component supports the sending of events. After an event source creates an event and populates it with data, the event source submits the event to an emitter. The emitter optionally performs automatic content completion and then validates the event to ensure that it conforms to the Common Base Event specification.

The emmiter also compares the event to configurable filter criteria. If the event is valid and passes the filter criteria, the emitter sends the event to the event service. An emitter can send events to the event service either synchronously, by using Enterprise JavaBeans™ (EJB™) calls, or asynchronously, by using a Java Message Service (JMS) queue.

Event service

The event service is the conduit between event sources and event consumers. The event service receives events that are submitted to emitters by event sources. It stores events in a persistent data store and then distributes them

asynchronously to subscribed event consumers. In addition, the event service supports synchronous queries of historical events from the persistent store.

Event catalog

The event catalog is a repository of event metadata. Applications use the event catalog to retrieve information about classes of events and their permitted content.

Event source

An event source is any application that uses an emitter to send events to the event service.

Event consumer

An event consumer is any application that receives events from the event service.

Event catalog application

An event catalog application is any application that stores or retrieves event metadata in the event catalog. This might be a management or development tool. It might also be an event source or event consumer.

Figure 1-1 shows the general flow of events from an event source to an event consumer by using the CEI.

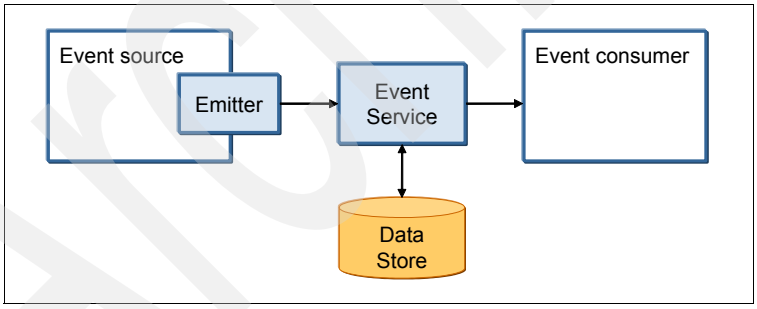


Figure 1-1 Event flow in the CEI

1.3 Common Base Event model

The Common Base Event model is a standard that defines a common representation of events intended for use by enterprise management and business applications. This standard, developed by the IBM Autonomic Computing Architecture Board, supports the encoding of logging, tracing,

management, and business events that use a common XML-based format, making it possible to correlate different types of events that originate from different applications. The Common Base Event model is part of the IBM Autonomic Computing Toolkit. For more information about autonomic comuting, see the Autonomic computing Web page at the following address:

http://www.ibm.com/autonomic

CEI currently supports version 1.0.1 of the specification.

The basic concept behind the Common Base Event model is the *situation*. A situation can be anything that happens anywhere in the computing infrastructure, such as a server shutdown, a disk-drive failure, or a failed user login. The Common Base Event model defines a set of standard situation types that accommodate most of the situations, such as StartSituation and CreateSituation, that might arise.

An *event* is a structured notification that reports information related to a situation. An event reports three kinds of information:

- The situation itself (what has happened)
- The identity of the affected component (for example, the server that has shut down)
- The identity of the component that is reporting the situation (which might be the same as the affected component)

The Common Base Event specification defines an event as an XML element that contains properties that provide all three kinds of information. These properties are encoded as attributes and subelements of the root element, CommonBaseEvent.

The Common Base Event format is extensible. In addition to the standard event properties, an event can also contain extended data elements, which are application-specific elements that can contain any kind of information relevant to the situation. The extensionName attribute labels an event with an optional classification name, an *event class*, which indicates to applications the sort of extended data elements to expect. The event catalog stores event definitions that describe these event classes and their allowed content.

For complete details about the Common Base Event format, see the specification document and XSD schema included in the IBM Autonomic Computing Toolkit.

Archived

Archived

2

WebSphere Business Monitor and concepts

In this chapter, we introduce IBM WebSphere Business Monitor. We describe the architecture of WebSphere Business Monitor V6.1 and define the associated terminology.

In addition, we discuss the new features in WebSphere Business Monitor V6.1. We demonstrate how WebSphere Business Monitor V6.1 extends capabilities and simplifies the way you monitor and manage the performance of your business.

Some of the information and figures in this chapter are from the article “What’s new in WebSphere Business Monitor 6.1,” which you can find at the following Web address:

http://www.ibm.com/developerworks/architecture/library/i-bam611/

© Copyright IBM Corp. 2008. All rights reserved.

2.1 Introduction to WebSphere Business Monitor

WebSphere Business Monitor is an integral part of the IBM business process management (BPM) portfolio. It is a comprehensive business activity monitoring (BAM) solution that provides a near real-time view of your business performance in the following ways:

- Provides visibility into the performance of business activities by processing events, calculating business metrics, and presenting key performance indicators (KPIs) through business dashboards

  Users can track current business performance against expectations and analyze trends over time.
- Helps when something goes wrong and in situations where expectations are not met
- Makes the organization aware of potential problems much earlier

  A directed action can be planned and carried out, taking the right action at the right time.

With WebSphere Business Monitor, you can understand business performance so that you can compare it with expected results. You can use the dashboards to see if KPIs are tracking to their desired targets and to determine if you have any unforeseen bottlenecks in your business process, such as in activities that involve human tasks. You can also use the actual historical results with other tools, such as WebSphere Business Modeler, to improve your business processes, for more accurate simulation of proposed changes.

WebSphere Business Monitor monitors your activities or processes by receiving and processing business events, called *Common Base Events*, from business applications. The events that the WebSphere Business Monitor server receives reflect your business activity. Information processed from events is stored in the WebSphere Business Monitor database.

Additionally, you can configure the WebSphere Business Monitor server to detect special business situations and manage the resulting actions. WebSphere Business Monitor can subscribe to business events from various sources. It can monitor business process applications running on WebSphere Application Server, WebSphere Process Server, and WebSphere Enterprise Service Bus. It can also monitor business process applications running on WebSphere MQ Workflow and FileNet P8 BPM. In fact, business events can be monitored from virtually anywhere if Common Base Events can be generated from their activities.

Archived

WebSphere Business Monitor also provides a development toolkit. This is an environment for technical users to create and test monitor models. Monitor models describe how to process events and how to collect information for use with dashboards.

In summary, to monitor your business operations, WebSphere Business Monitor offers the following functions:

- Captures business-related data that is specified by your requests from business applications based on the monitor model that you define and install
- Extracts the measurement variables from the data
- Transforms the variables into metric and KPI values
- Displays the measurement values on your dashboards
- Provides business intelligence insight through dimensional analysis and reporting
- Enables you to define actions to take when specified situations occur
- Identifies and notifies you of operation failures for inspection and analysis

2.2 WebSphere Business Monitor V6.1 architecture

Figure 2-1 on page 14 shows an overview of how WebSphere Business Monitor monitors applications. When we receive the user requirement for business monitoring, we use the WebSphere Business Monitor development toolkit, that ships with WebSphere Business Monitor, to develop a monitor model. In this model, we choose the interested business events and define the business measures and situations. Optionally, we can create an initial model by using the WebSphere Business Modeler. Then we achieve a deployable monitor model application.

Next, we deploy the monitor model to the server in which the monitor model executes and processes the events emitted by the monitored business process or other application. The WebSphere Business Monitor dashboards provide clients with a number of views for different types of data representation.

Archived

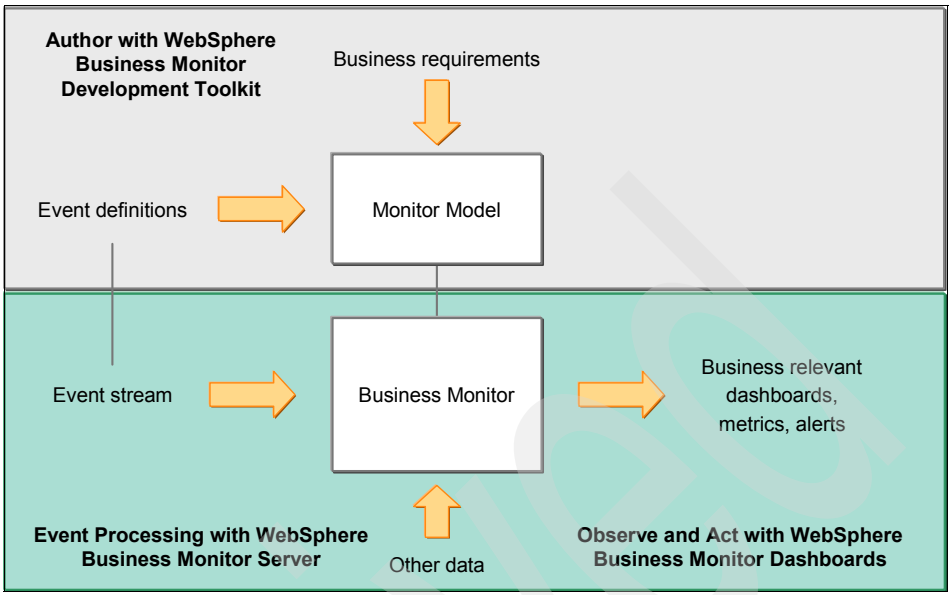


Figure 2-1 How WebSphere Business Monitor monitors applications

In the following sections, we introduce the WebSphere Business Monitor buildtime and runtime architecture and concepts respectively.

2.2.1 WebSphere Business Monitor V6.1 buildtime architecture

Figure 2-2 on page 15 shows the WebSphere Business Monitor V6.1 buildtime architecture. This architecture consists of the following components:

- The *WebSphere Business Monitor development toolkit* is installed upon Rational® Application Developer or WebSphere Integration Developer, which introduces a new Business Monitoring perspective.
- With the WebSphere Business Monitor development toolkit, you can develop a monitor model in the *Monitor Model editor*. You can load the event definitions that describe the types of events that will be emitted by the monitored applications.
- WebSphere Business Monitor development toolkit also provides the *Monitor test environment* with which you can publish your monitor model and unit test by emitting events from monitored applications and verifying the results from the Web dashboard.

- *WebSphere Business Monitor Integrated Test Client* helps to emit sample events. It is installed as part of the WebSphere Business Monitor development toolkit installation. By using this tool, you can quickly and conveniently test a monitor model.

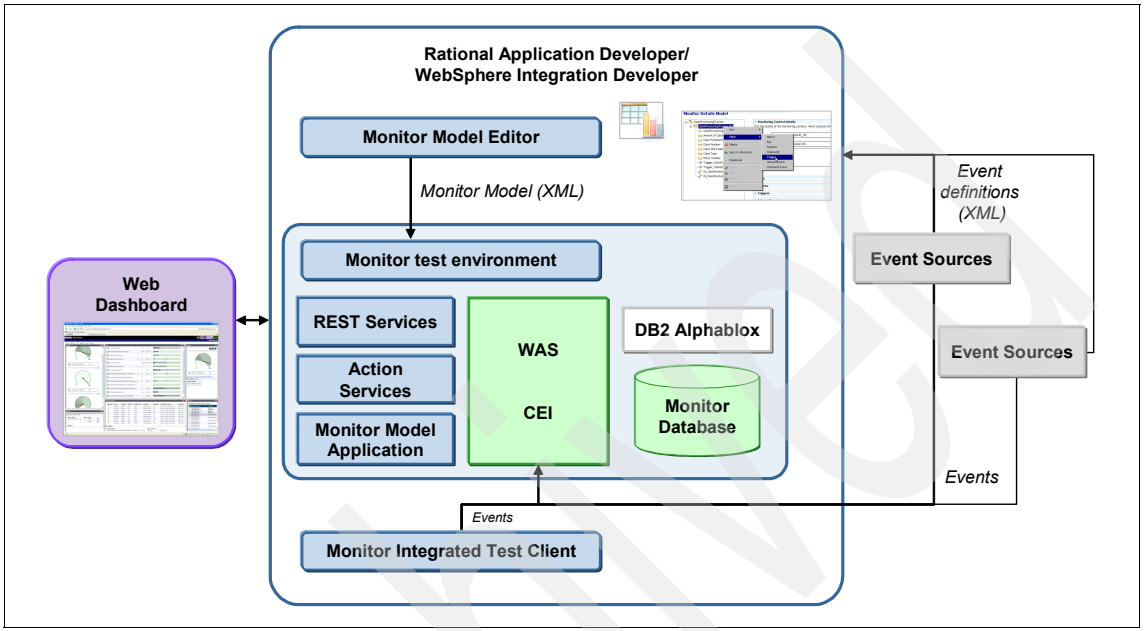


Figure 2-2 WebSphere Business Monitor V6.1 buildtime architecture

In the following sections, we explain the iterative development steps for the monitor model and introduce each part of the WebSphere Business Monitor V6.1 buildtime architecture.

Archive

Iterative development steps

The WebSphere Business Monitor development toolkit provides a buildtime environment in which you can achieve iterative development on a monitor model. Figure 2-3 illustrates the general steps, which are as follows:

1. Define the monitor model within the Monitor Model editor.
2. Generate a J2EE™ project based on the developed monitor model.
3. Choose a test environment to deploy the model
4. Publish the tested monitor model to the test environment.
5. Run test events by executing the monitored application or Integrated Test Client, and check the monitoring results through the Web dashboard.
6. Return to the Monitor Model editor and refine the model.

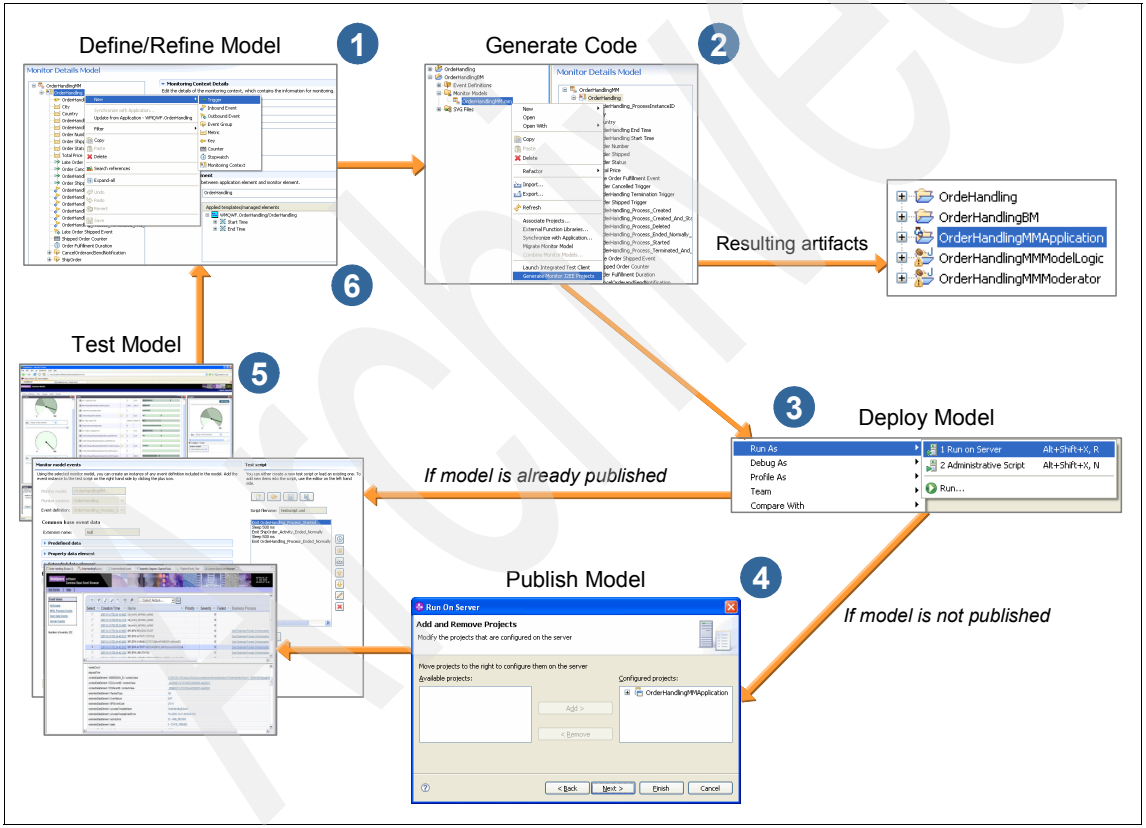


Figure 2-3 Iterative development on a monitor model

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 an initial model generated from WebSphere Integration Developer or WebSphere Business Modeler, or it can be created from scratch using the Monitor Model editor. The Monitor Model editor exposes the Monitoring Programming Model with a graphical user interface (GUI) to aid the user in their development effort (Figure 2-4).

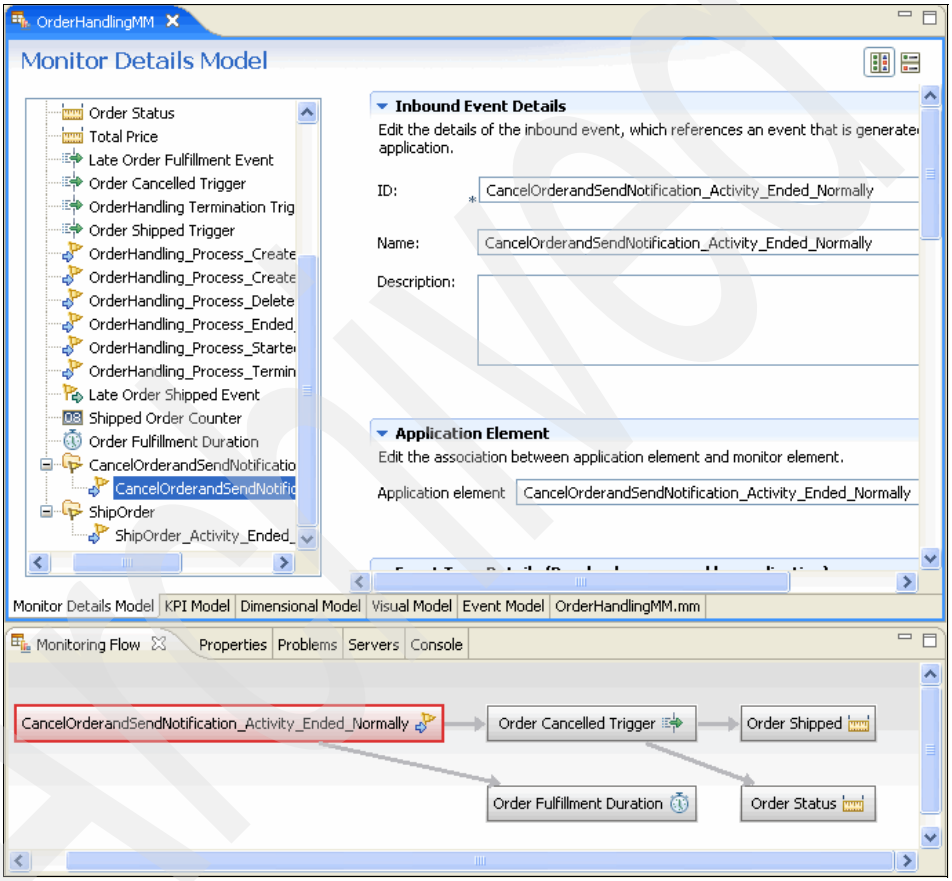


Figure 2-4 Monitor Model editor

Monitor model terminology

A monitor model consists of several parts that describe different aspects: the monitor details model, the KPI model, the dimensional model, the visual model, and the event model. In this section, we introduce the parts respectively and define terms that are associated within them.

Monitor details model

The monitor details model is a container for monitoring contexts and their associated metrics, keys, counters, stopwatches, triggers, and events. The monitor details model holds most of the monitor model information, which is also required by the other models to perform further analysis. Table 2-1 describes the components of the monitor details model.

Table 2-1 Monitor details model

Monitor details model	Description
Monitoring context	A monitoring context defines all of the data that should be collected about an entity, such as a process, customer order, or the stock level of an item in a warehouse, while the system is running. Each of its runtime instances (monitoring context instances) use incoming events to monitor a particular (real or abstract) entity, such as a particular process execution, the state of a particular order, or the stock level of an item in a warehouse. They collect information that, individually or in combination, is useful for making business decisions. The information is extracted from the data carried by inbound events and is held in metrics, counters, and stopwatches, which represent the business measures that a monitoring context collects.
Event group	Event groups are containers for inbound events that enable you to group related inbound events together without using a monitoring context. Monitoring contexts introduce overhead in the form of keys, cubes, and so on. Event groups are a visual construct and are not represented in the monitor model XML file.
Inbound event	A monitoring-enabled application generates a series of events. To indicate the events that are of interest to the monitoring context, you define inbound events in the Monitor Model editor. These are subscription points or entry points for inbound events in a monitoring context. WebSphere Business Monitor subscribes to the events that you specify and delivers them to all event entry points with a matching event subscription.
Metric	A metric is a holder for information, usually a business performance measurement, in a monitoring context. 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. Examples of metrics are the working duration of a process, the name of the user assigned to a task, a supplier's response time, and the cost of the risk assessment step in an insurance process.

Monitor details model	Description
Key	A key is a piece of information that characterizes and identifies the real-world entity that is being tracked by a monitoring context. Every monitoring context must have one or more keys. The set of these keys, together with the keys of any ancestor monitoring contexts (parent, parent's parent, and so on) are combined to uniquely define the real-world object.
Stopwatch	KPIs often depend on elapsed time. You can define stopwatches to keep track of elapsed time, which can be started, stopped, or reset by a trigger or an inbound event.
Counter	KPIs often depend on the number of occurrences of a situation or event. You can define counters to count such occurrences, which can be incremented, decremented, or reset to zero by a trigger or an inbound event.
Trigger	A trigger is a mechanism that detects an occurrence and can cause additional processing in response. For example, you can define a trigger that causes a metric to be updated, a counter to be incremented, or a stopwatch to be halted each time a task ends. A trigger can also cause an expression to populate an outbound event, which subsequently is sent to the Common Event Infrastructure (CEI).
Outbound event	Outbound events can be emitted from a monitoring context. They can be received by any event-processing application. One of these services is the *Monitor action services*, with which an administrator can specify the actions to take in response to an event. The receiving application can also be WebSphere Business Monitor, so that an outbound event from one monitoring context can be an inbound event to another monitoring context or to a KPI context.

Archive

Figure 2-5 shows the flow of control between the monitor details model elements. For example, an incoming event might fire a trigger, which in turn, controls a stopwatch. It might also increment a counter and provide input to a map that updates a metric.

Some elements, such as trigger and map definitions, carry expressions. For example, trigger conditions control whether a trigger fires, and map expressions define a map's output as a function of its inputs. These expressions can depend on fields that are defined in the monitoring context (metrics, counters, and stopwatches) as well as on event attributes if their evaluation is caused by an inbound event.

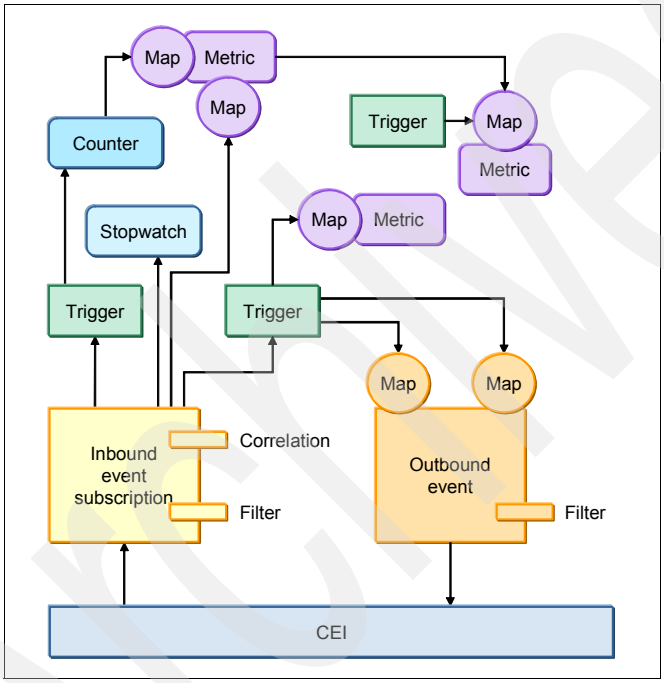


Figure 2-5 Monitoring context

Archive

KPI model

The KPI model is the part of the monitor model that contains the KPI contexts, which in turn contain KPIs and their associated triggers and events. KPIs use aggregate data, such as an average or sum of the values of a particular metric, which can be qualified by a time period or other filter. Table 2-2 describes the components of the KPI model.

Table 2-2 KPI model

KPI model	Description
KPI context	A KPI context is a container for KPIs and their associated triggers and events. Unlike a monitoring context, a KPI context has no keys or metrics. You must create a KPI context as a container before you can create KPIs.
KPI	KPIs are quantifiable measurements of the improvement or deterioration in the performance of an activity critical to the success of a business. These measurements break down key areas of your business so that you can see how they contribute to business results. KPIs differ depending on the business. For example, in a call center, the timely answering of customer calls is a key business activity. A KPI can be the average time for response to a customer call for the last 30 days. This KPI can have a target of less than one minute.

Dimensional model

The dimensional model is the part of the monitor model that defines the cubes that are used for dimensional analysis, which can provide a business intelligence perspective of the historical data collected across all instances. You must define the dimensional model if you plan to install IBM DB2® Alphablox and perform dimensional analysis in the dashboards.

Table 2-3 describes the components of the dimensional model.

Table 2-3 Dimensional model

Dimensional model	Description
Cube	A cube is a multidimensional representation of the information needed for online analytical processing. The dimensional model overlays a cube structure on a monitoring context definition, which defines the way that information collected by monitoring context instances is stored. To perform dimensional analysis in the dashboards, you must install IBM DB2 Alphablox with WebSphere Business Monitor.
Dimension	Dimensions are data categories that are used to organize and select monitoring context instances for reporting and analysis. Examples of dimensions are time, accounts, products, and markets. Dimensions are composed of one or more hierarchical levels. For example, a Location dimension can have City, Region, and Country levels.

Dimensional model	Description
Measure	Measures are calculations based on metrics, keys, counters, or stopwatches. A measure points to a metric (such as order amount) and specifies an aggregation function (such as average or sum). You can define measures to be used in dimensional analysis in the dashboards.

Visual model

The visual model is the part of the monitor model that contains visualizations. Each visualization contains a reference to a Scalable Vector Graphics (SVG) diagram that can be displayed in the WebSphere Business Monitor dashboards. In addition, each visualization can include a set of actions that describes how and when the diagram will be modified based on the values of metrics and KPIs.

Visualizations for each monitoring context and each KPI context consist of the following parts:

- A reference to an SVG file stored externally to the monitor model
- A group of named shape sets that provide a high-level mechanism for referencing sets of individual shapes in the SVG diagram
- Zero or more actions that define the visual transformation to perform on the SVG elements identified by the shape sets

Event model

The event model is the part of the monitor model that contains references to all of the elements of the event definitions used in the monitor model. The event model references each event definition that you use as an inbound or outbound event type in the monitor details model or KPI model. It also includes references to any schemas that are used to describe the structure of individual event parts.

WebSphere Business Monitor test environment

The WebSphere Business Monitor development toolkit provides a unified development and test environment for monitor model development. By using the WebSphere Business Monitor test environment, you can test the newly developed model that involves checking the values of metrics, KPIs, gauges, and so on. Monitor model testing involves verifying that the intended model logic is run correctly. Verification is accomplished by checking the results displayed in a dashboard.

Archive

Integrated test client

The integrated test client (Figure 2-6) is installed as part of the WebSphere Business Monitor development toolkit installation and is integrated into Rational Application Developer or WebSphere Integration Developer. You can use this tool to emit sample events. By using this tool, you can quickly and conveniently test a monitor model that is published in the Monitor test environment.

The Integrated Test Client has two sections. The section on the left side of the window is the *monitor model events editor*. In this section you can select, modify, and add events from the monitor model to the test script. The section on the right side is the *test script editor*. In this section you can add and edit commands, rearrange events and commands, and import Common Base Event files into the test script. When using the Integrated Test Client, you can create and emit sample events for inbound events only. The Integrated Test Client can handle events from both WebSphere Business Monitor version 6.0.2 or 6.1.

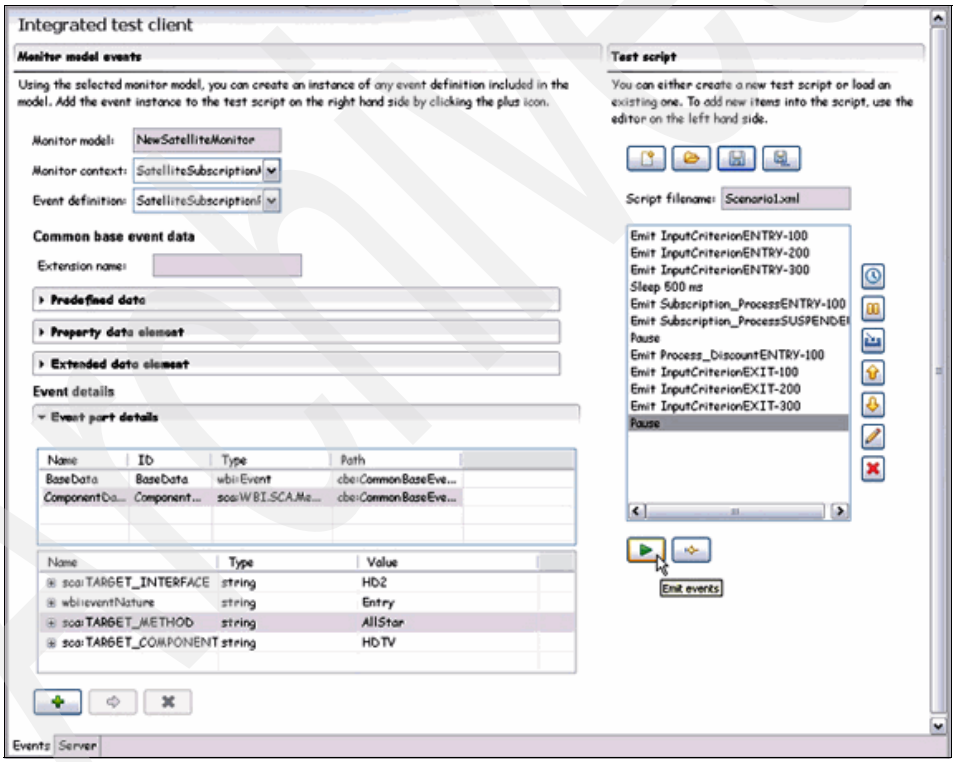


Figure 2-6 Integrated Test Client

2.2.2 WebSphere Business Monitor V6.1 runtime architecture

WebSphere Business Monitor V6.1 provides two types of dashboards: Web-based dashboards and portlet-based dashboards. *Web-based dashboards* are implemented as Web pages. *Portlet-based dashboards* are a component of WebSphere Business Monitor V6.1 that operates within the WebSphere Portal V6.0.1.1 environment. Figure 2-7 on page 25 and Figure 2-8 on page 26 illustrate the runtime architecture respectively. As indicated by the flow in both figures, you follow this sequence:

1. After you finish the development of the monitor model, export the resulting monitor application that is deployed into the runtime environment (WebSphere Application Server, WebSphere Enterprise Service Bus, or WebSphere Process Server) by using the WebSphere Administrative Console. Ensure that you configure the WebSphere Business Monitor server application to link appropriately to CEI to be registered to consume events from the emitting application.
2. On another server (for example, WebSphere Process Server or WebSphere MQ Workflow), deploy the application that emits events to a CEI server.
3. Upon receipt of the registered event types, the CEI server sends the appropriate Common Base Event to the server registered in step 1.
4. The monitor model executes. It polls for events waiting on the queue, processes them according to the monitor model instructions, and stores business measures to the Monitor database.
5. Optionally, the monitor model application can obtain real-time event-related data from the event or data source through code that you can provide and invoke as a user-defined XPath function.
6. Optionally, in addition to receiving events as in step 3, the monitor model application can detect business situations (for example, thresholds) and emit events that can be consumed by the WebSphere Business Monitor action services.
7. The CEI server routes events received through the event or data source or the monitor model application to the action services.
8. The action services component takes action such as sending notifications, calling Web services, and invoking Service Component Architecture (SCA) components to perform actions.

Archive

9. Independently, the business user uses the dashboards, which invoke Representational State Transfer (REST) services to return data from the Monitor database and show monitoring results through different views.
   – Figure 2-7 shows usage of the lightweight Web dashboard, which is implemented as Web pages. DB2 Alphablox package installation is optional and is only the prerequisite for use of the dashboard's dimensional and report views.

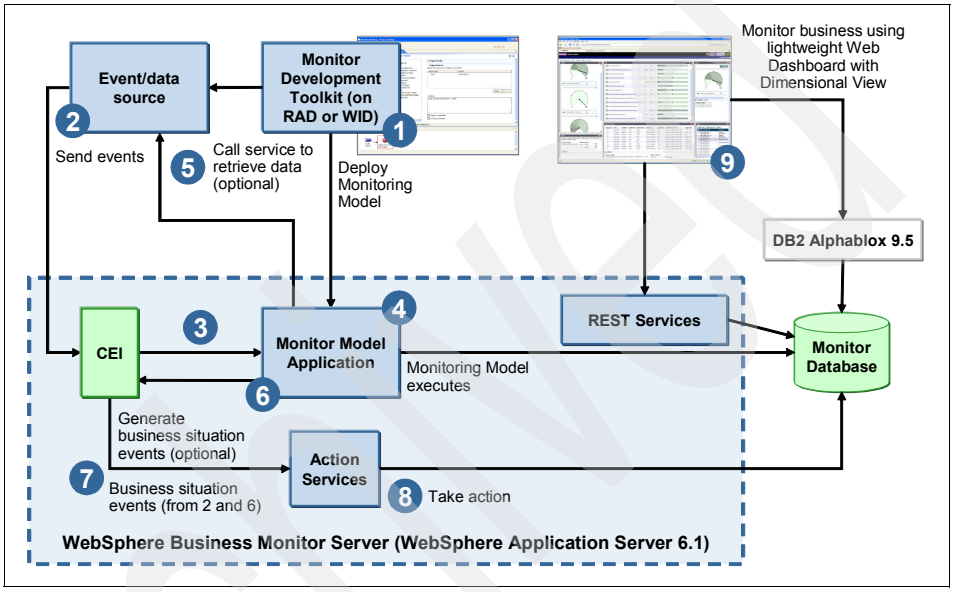


Figure 2-7 WebSphere Business Monitor V6.1 run time with Web dashboard

Archived

– Figure 2-8 shows usage of the Portal dashboard, which uses WebSphere Portal Server to invoke a REST service. DB2 Alphablox package installation is optional and is only the prerequisite for use of the dashboard's dimensional and report views.

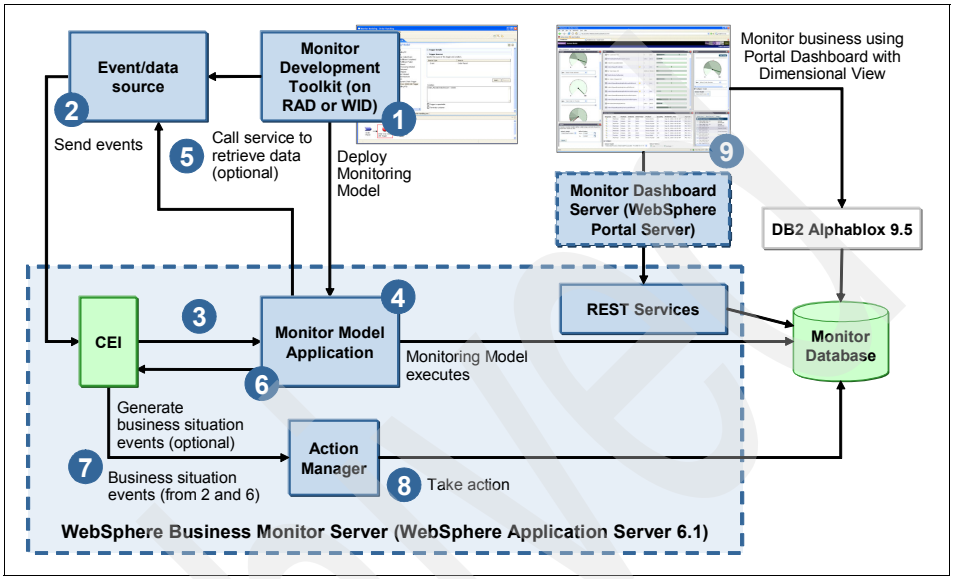


Figure 2-8 WebSphere Business Monitor V6.1 run time with Portal dashboard

In the following sections, we introduce the main parts of the WebSphere Business Monitor V6.1 runtime architecture.

WebSphere Business Monitor server

The WebSphere Business Monitor server executes the monitor model to extract the necessary information from the monitored process events, stores the information into a repository for later analysis, and responds to business situations. It consists of the WebSphere Business Monitor model manager, the WebSphere Business Monitor action services, REST APIs, and data movement service.

Monitor model manager

The monitor model manager acts as the execution engine for the monitor model that is created by using the WebSphere Business Monitor development toolkit. During model development, you can define the metrics and KPIs to monitor and the events used in calculating them. The monitor model manager then uses the model, receives the events, and calculates the metrics to be persisted for future display on dashboards.

The monitor model manager handles all of the metric calculations and business situation detections that are required for business process management. It creates, reads, updates, and terminates context instances. It also calculates the metric values of these instances.

The monitor model manager consumes events from the CEI to update the metrics and KPIs displayed on dashboards. It also sends events to the CEI, as defined by the monitor model.

Action services

WebSphere Business Monitor action services is a component of WebSphere Business Monitor that invokes action services when it receives defined situation events emitted by WebSphere Business Monitor and other applications. The actions are based on user-defined action templates. Action types include notifications, calling Web services, and invoking SCA components.

REST APIs

The REST APIs are public APIs that you can use to extract monitor data or to create custom dashboards and reports based on monitor data. The REST implementation allows for easy to use HTTP services that are language- and platform-independent, stateless, scalable, and easily parsed. The REST APIs are extremely useful for creating Ajax-style Web applications, in the same vein as the WebSphere Business Monitor dashboards.

Data movement services

Data movement services optimize server processing and reporting in higher volume production environments. Operational tables are optimized for inserts and updates, and the reporting tables are optimized for dashboard queries. When enabled, the data movement service runs automatically as a WebSphere application. It is configured and monitored on the WebSphere Application Server administrative console.

Data movement service replaces DB2 Replication Manager from previous releases and works across DB2 and Oracle®. When enabled, the data movement service cannot be disabled. The data movement service is optional in this release. We do not recommend the data movement service for development and test environments or small production environments where performance is not a concern. Therefore, it is not supported in the WebSphere Business Monitor test environment.

Monitor database

WebSphere Business Monitor V6.1 is composed of a single database, usually called *Monitor*. It stores information about the current state of monitoring contexts to be used for event processing by the Business Monitor server

component. It also stores definitions of the deployed monitor models along with other WebSphere Business Monitor information and the corresponding KPI and metric definitions and provides the data for dashboards.

The Monitor database stores both instances (runtime) and metadata related to a model. The runtime data consists of information about running instances. This information includes metrics as well as the final status of recently completed or failed processes. The metadata describes information about deployed monitor models. This metadata is not moved by the data movement service.

Each monitoring context is represented as a table in the Monitor database. These tables can be automatically created as part of the monitor model installation or created later. The tables must exist before the WebSphere Business Monitor server can process events for the monitor model. These same tables are used to support the dashboards. Under high volumes, there can be contention between the WebSphere Business Monitor server and dashboard's use of these tables. In these cases, the data movement service should be enabled. This creates a duplicate set of tables dedicated for dashboard queries and initiates the transfer of data from the server tables to the dashboard tables.

WebSphere Business Monitor dashboards

BAM solutions are usually deployed to give business users more visibility and insight into the performance of the business and to help them take informed action. While BAM can sometimes cause actions to be taken automatically, the principal user interfaces for monitoring and action are the WebSphere Business Monitor dashboards.

Two kinds of dashboard are provided: Web dashboard and Portal dashboard. Both types of dashboard have the following views:

Archive

- Instances

  The Instances view (Figure 2-9) shows the details of activity and process instances, either individually or in groups. You can view both active and completed instances by using different metric filters. You can use a time metric as a filter to control instance data to be displayed for a specific time interval.

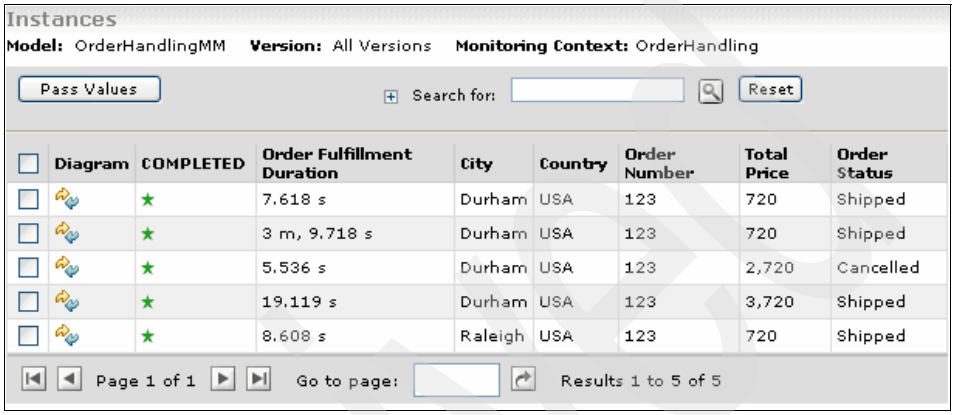


Figure 2-9 Instances view

- KPIs

  The KPIs view (Figure 2-10) shows the values of individual KPIs compared to the ranges and targets that you have defined for your business. KPIs can be viewed by using various forms of gauges, bar graphs, and tables.

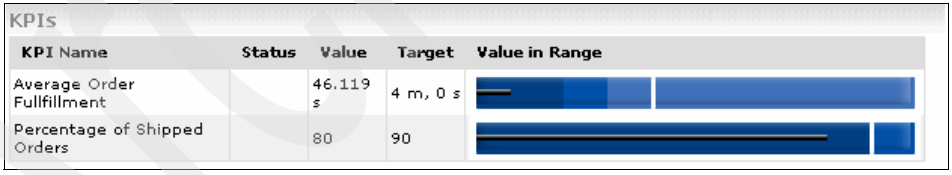


Figure 2-10 KPIs view

- Diagrams

  The Diagrams view (Figure 2-11) shows a graphical depiction of flows and the status of activities, such as a map or a process flow model. You can create virtually any visual diagram and add metrics and KPI values. You can also visually change diagram shapes as events are processed.

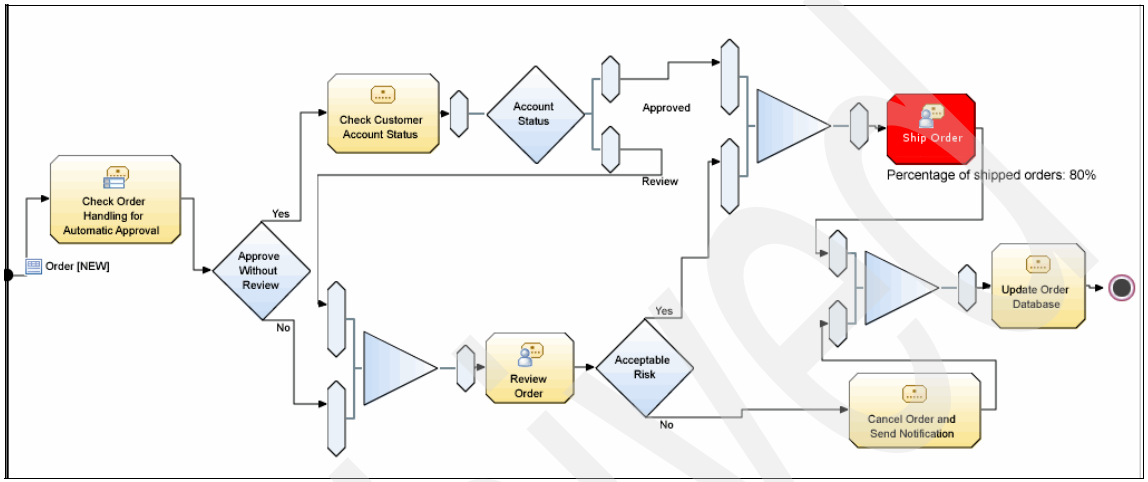


Figure 2-11 Diagrams view

- Alerts

  With the Alerts view (Figure 2-12), you can view or sends notifications to a specific user when business situations occur.

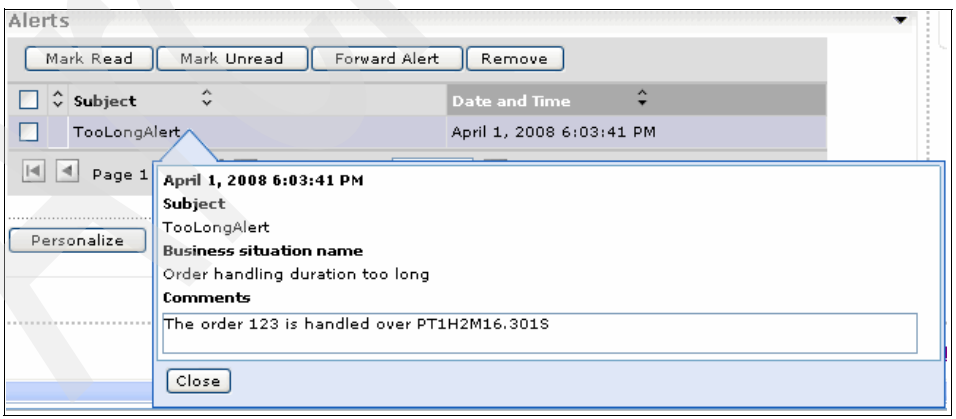


Figure 2-12 Alerts view

- Reports

  The Reports view (Figure 2-13) shows tables and graphs with analysis of performance across dimensions of your business relative to time periods (quarter, months, days, or hours).

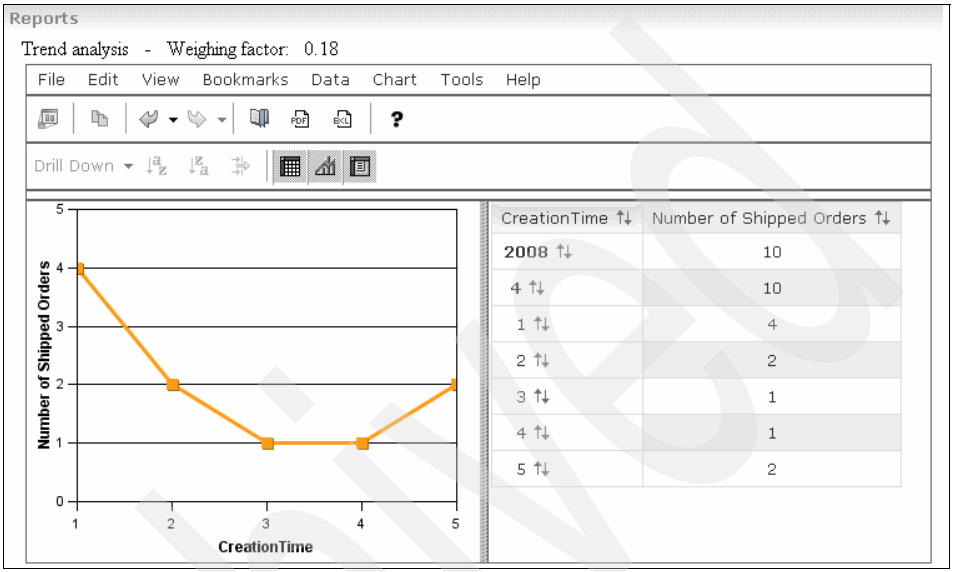


Figure 2-13 Reports view

Archived

- Dimensions

  The Dimensions view (Figure 2-14) provides granular details that show how critical aspects of your business are performing. You can drill up or down from a dimension to understand a specific aspect of your business performance.

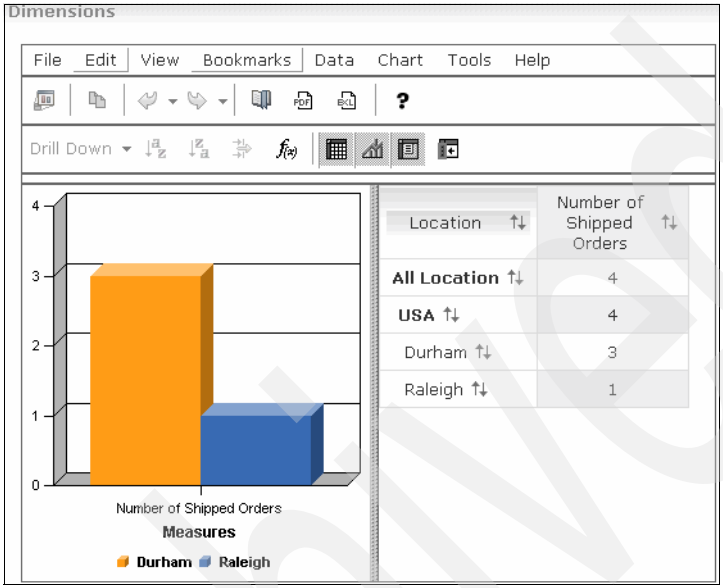


Figure 2-14 Dimensions view

Archived

- Human Tasks

  The Human Tasks view (Figure 2-15) provides metrics for work that people are performing within a process, helping to identify bottlenecks and enabling actions such as rebalancing workload. From this view, you can also perform operations on the selected human tasks, such as claim a task or transfer a task to another user.

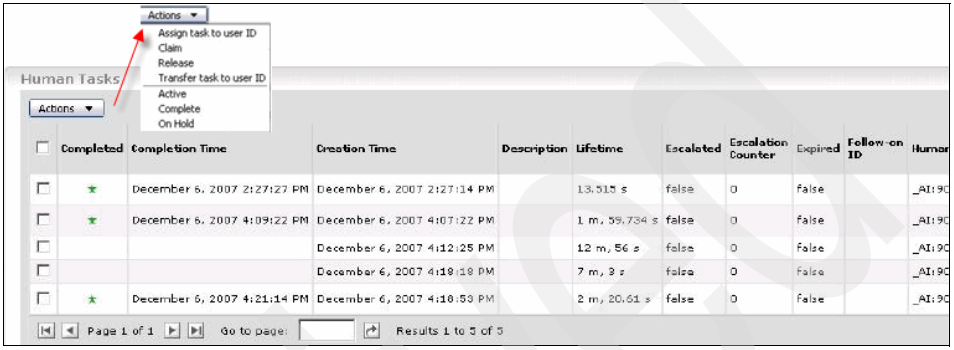


Figure 2-15 Human tasks view

In addition to these view, the Portal dashboard has the Organizations view. This view displays the structure of the organization (such as, organization units and employees) defined in the user registry, for example, a Lightweight Directory Access Protocol (LDAP) server. This item is used in conjunction with Reports to perform data analysis for assigned employees.

Three WebSphere Business Monitor utilities are supported and can be accessed through the WebSphere Business Monitor dashboards:

- Alerts Subscription

  Use this utility to subscribe and unsubscribe to different predefined alerts. Along with subscribing to an alert, the user can choose the type of notification channel associated with each alert.

- Export Values

  Use this utility to export data resulting from a specific monitor model and duration to an XML file that can be imported by WebSphere Business Modeler.

- KPI Manager

  Use this utility to define, copy, and update KPIs directly from the dashboard interface.

2.3 New in WebSphere Business Monitor V6.1

In WebSphere Business Monitor V6.1, several new features extend its capabilities and make it much easier to use. In the following sections, we explain iterative development, the more flexible architecture for monitoring events, the highlights of the business user experience in Web 2.0 dashboards, and simplified installation and administration.

2.3.1 Development toolkit

The benefits from BAM are realized as business users gain insight and take actions to respond to opportunities and improve. Any BAM project must be enabled through activities, such as project requirements gathering, design and development of monitor models, and testing and deploying the models. In WebSphere Business Monitor V6.1, the WebSphere Business Monitor development toolkit brings major enhancements to speed the time to value across these project activities.

Authoring WebSphere Business Monitor models

With WebSphere Business Monitor V6.1, authoring a monitor model has been greatly simplified, with broader editor support, reduced steps for common tasks, and improved generation capability. We highlight only the following three of many authoring improvements in WebSphere Business Monitor V6.1:

- Monitor model templates
- Refactoring and synchronization features
- Visual Model editor

Because WebSphere Business Monitor is often used in combination with WebSphere Integration Developer, WebSphere Enterprise Service Bus and WebSphere Process Server, WebSphere Business Monitor V6.1 provides a much deeper level of integration. Perhaps the first noticeable change is the addition of *monitor model templates* for common monitoring scenarios. The templates can be applied to such WebSphere Integration Developer components as Business Process Execution Language (BPEL) processes, human tasks, and WebSphere Enterprise Service Bus mediation modules.

Archived

Figure 2-16 shows an example of templates being applied for monitoring a BPEL process. You can choose to add metrics and KPIs to calculate the durations of processing, and to track the current state of processes and activities.

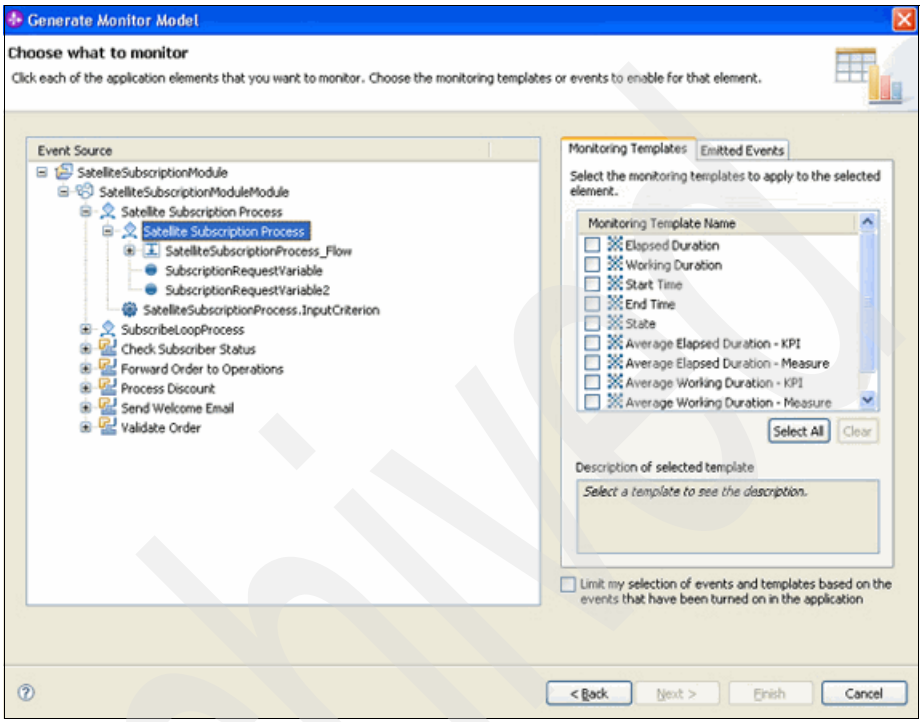


Figure 2-16 WebSphere Business Monitor model generation

When a developer or team of developers are using WebSphere Integration Developer to simultaneously create an application and a monitor model for that application, WebSphere Business Monitor 6.1 offers new *refactoring and synchronization features*. Many changes made to the WebSphere Integration Developer application trigger corresponding monitor model changes to be made.

The *Visual Model editor* is another new feature in the WebSphere Business Monitor development toolkit. With this editor, you define how a diagram is associated with a set of shapes and a set of actions, describing how and when the diagram will be modified based on the values of metrics or KPIs.

Figure 2-17 shows an example of a developer using the Visual Model editor to unit test the display with a sample data set. After making changes to the visual model, you can add sample data to test the diagrams from right inside the WebSphere Business Monitor toolkit. You can also check the return values of visual model expressions to make sure that they are working as intended.

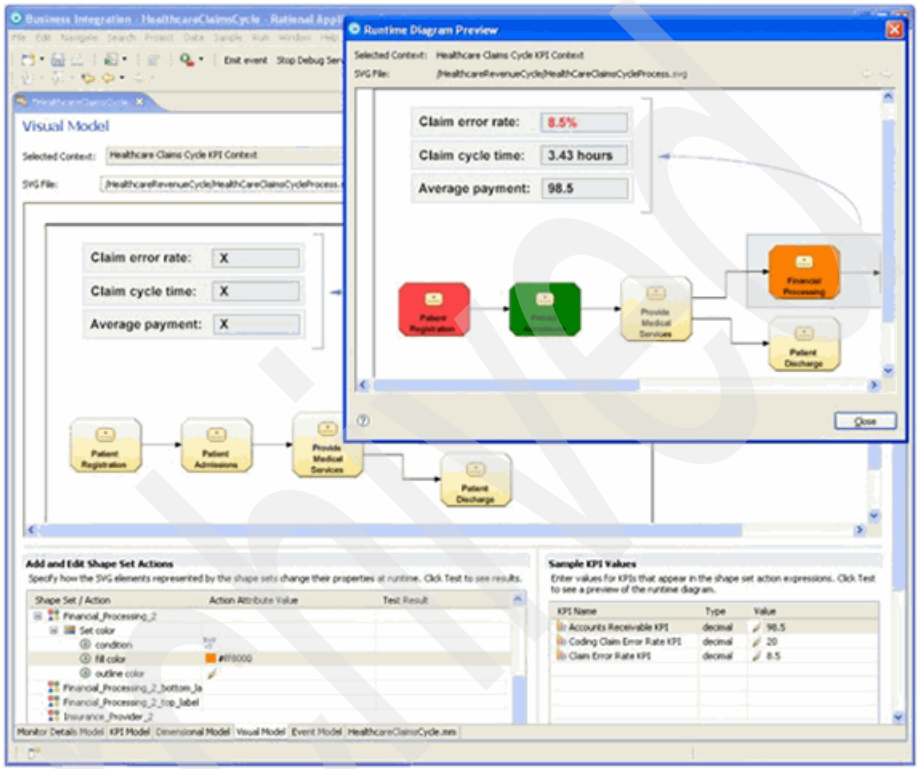


Figure 2-17 Visual Model editor

Unit testing with an integrated WebSphere Business Monitor server

The WebSphere Business Monitor development toolkit also offers the following tools:

- Unit test server
- Integrated test client
- Integrated dashboards

Unit test server

The WebSphere Business Monitor development toolkit has more than an editor for the monitor model XML file. It provides a unit test environment in which you

can test your monitor model in an embedded server environment. With this testing, you can see if your monitor model properly reacts to events by storing the right data and taking the right actions.

The unit test server is a full WebSphere Business Monitor server, running in a specially created profile that has been marked as `developmentMode=true` in its server.xml file. It offers certain “fast-path” options, such as autocreation of tables and autoregistration of cubes. This mode cleans potentially invalid data from previous iterations, thus avoiding the complexities associated with a multi-version environment. During iterative development, you want to quickly try changes and move on, without involving a database administrator or a WebSphere administrator.

The unit test server, which uses the Derby Embedded database that is a part of WebSphere, is controlled by the standard mechanisms as provided in Rational Application Developer V7 or WebSphere Integration Developer V6.1. (You do not need a separately installed database in order to do basic testing of your monitor model.) The Add/Remove Projects wizard, the ability to run projects from the workspace, and ability to republish such projects as you make changes are included.

Integrated test client

An important new feature is the ability to compose and send events from within the integrated development environment (IDE). You can test that your monitor model correctly processes events, without executing your application that is being monitored. A basic form-based graphical interface lets you fill in values for each field of the event according to its XML Schema Definition (XSD). It enforces that data of the proper type is entered for each field and offers assistance such as a pop-up calendar for fields of type dateTime.

As shown in Figure 2-6 on page 23, with the integrated test client, you can build up scripts of events (including pauses), export and import such scripts, and re-order and re-edit the events within the script. It is easy to build libraries of event data that can be easily resubmitted, such as when testing a change to a monitor model.

Integrated dashboards

The Integrated dashboards are a critical new feature that provides full support for all dashboard views within the embedded browser of the IDE by using the new non-Portal-based dashboard option. Just as you can launch the administrative console or the Common Base Event browser from the pop-up menu for the unit test server, you can also launch the Web dashboards to run within a page in the IDE. You have the full capabilities of the dashboards, as when used in a production environment, including the dimensional analysis views, annotated SVG diagrams, alerts, and all the other capabilities of the WebSphere Business

Monitor dashboards. During iterative development, you can quickly make changes to your monitor model, republish the results to the unit test server, and immediately see the results in the dashboards.

Developing custom BAM interfaces

WebSphere Business Monitor provides several dashboard views you can use for a wide range of business dashboard requirements. The views can be personalized for appearance and behavior. In some cases, an organization might have specialized requirements for accessing and using the BAM data that was processed and calculated by the WebSphere Business Monitor server. To better support these cases, WebSphere Business Monitor V6.1 offers two new ways to access BAM data.

One new access mechanism is based completely on a Web 2.0 approach. WebSphere Business Monitor V6.1 publishes a set of REST services for accessing the metadata and data values. Consistent with the REST architecture pattern, each service is identified through a URI. For example, to retrieve a list of KPIs, you can use the URI in the form of /models/model_ID/versions/version_ID/kpis. These REST services use the data interchange format JavaScript™ Object Notation (JSON) for representing the payload of information exchanged.

Example 2-1 shows the JSON object fields returned from the request for the list of KPIs.

Example 2-1 List of KPIs

```
[ --- KPI array
   { -- 1ST KPI
     "KPI ID":  "kpi id1",
     "Model ID": "model1",
     "Version": "1234556",
     "KPI Display Name": "kpi name",
     "KPI Origin": "modeled",
     "User ID": "user 1"
     "KPI Context ID": "context id1"
     "KPI Description": "sample KPI Data"
     "KPI Data Type": "duration"
     "Target": 123344
     "Target Localized": "xxxxxxxxxxx"
     "KPI Calc Method": "aggregated"
     "Aggregated Metric ID": "metric ID"
     "Aggregated Metric Name": "metric name 1"
     "Aggregated Function": "avg"
     "Version Aggregation": "allVersions"
```

```
    "View Access": "public"
  },
  { -- 2nd KPI
  }
]
```

A second new way to access BAM data from WebSphere Business Monitor is a point of integration with a complementary product called *WebSphere Portlet Factory*. The WebSphere Business Monitor Data Access Builder for Portlet Factory is available as a Web download from the IBM SOA Business Catalog.

With the Data Access Builder, using Portlet Factory, you can easily retrieve WebSphere Business Monitor metadata and data values. After the builder is installed in the Portlet Factory development tool, you can combine the range of Factory visual builders with the data from WebSphere Business Monitor. For example, you can create a custom portlet to visualize a KPI calculated by WebSphere Business Monitor and deploy that portlet alongside one of the portlets installed by WebSphere Business Monitor. You can also use Data Access Builder with the WebSphere Dashboard Framework, which is a super-set of Portlet Factory, adding hi-fidelity charting to your toolbox for creating customer interfaces.

2.3.2 Monitor server

The more activities that are available to observe and monitor, the more value you gain from BAM. WebSphere Business Monitor V6.1 provides new ways to access more events and data, making KPIs and dashboards a more comprehensive representation of business performance. We discuss the changes in WebSphere Business Monitor V6.1 that are relevant to four categories of business activity to be monitored. These categories are often employed in combination.

WebSphere Business Monitor V6.1 brings two fundamental changes that lead to great flexibility in event processing:

- XML best practices for event schema and runtime content

  Prior to V6.1, WebSphere Business Monitor required that events be described by using the CEI Event catalog format. This XML document is useful for describing event structures and field types, with the specification of required and optional fields. The catalog XML was functional for the purpose, but it required you to use a special XML dialect for an event schema.

  WebSphere Business Monitor V6.1 offers you a choice of using XSD for an event schema, which has many benefits. Events can now be described by using the same tools used for other aspects of service-oriented architecture (SOA) and BPM. If you have defined XSD for business data that you want to

pass in the body of a business event, you can simply reuse that XSD for purposes of business monitoring.

For example, if you authored a BPEL application in WebSphere Integration Developer, with a business object representing a customer order, you can reuse that business object definition in your monitor model. You no longer have to keep two different format definitions in sync. You have one format for both purposes. This also applies in cases where the customer uses WebSphere Service Registry and Repository to enforce governance around business object definitions, for example. WebSphere Service Registry and Repository has native support for XSD. Its Eclipse plug-ins can be used in combination with the WebSphere Business Monitor V6.1 development toolkit to extract such XSDs from the repository and use as both inbound and outbound event definitions when authoring monitor models.

WebSphere Business Monitor V6.1 also allows XML document fragments to be used as the body or payload of the business event. Previously, events delivered at run time were required to be *shredded* into parent and child hierarchies as extendedDataElements, often requiring a special transformation step before the event could be emitted by the source application. With WebSphere Business Monitor V6.1, business event payload data that is in an XML document form can be placed into the business event without requiring shredding. This flexibility provides an easier-to-use option for applications and middleware that can use XML.

- Support for event sequencing

  Asynchronous processing and network latencies can lead to cases where events arrive to WebSphere Business Monitor in an order that does not match the original sequence of business activity processing. For example, a simple order activity may have two events: one to signify order placement and another to signify order fulfillment. In some cases, the order fulfillment event can arrive before the order placement event. Prior to WebSphere Business Monitor V6.1, the monitor model developer and administrator were required to define emission of events to be ordered, usually by setting emission to be synchronous.

  WebSphere Business Monitor V6.1 introduces the capability to moderate the intake of events, performing a level of sequencing before the events are processed by monitor models. Support for event sequencing creates many new options for event emission. Since events can be now be emitted asynchronously, event sources that benefit from asynchronous processing (for example, WebSphere Enterprise Service Bus) become natural sources for business events. Event producers now have the option of putting events directly onto the input JMS queue for CEI, including with queue connections from alternate JMS providers such as WebSphere MQ. In addition, administrators now have such options as clustering CEI and its messaging engine if desired, even with such asynchronously emitted events.

Monitor business processes

WebSphere Business Monitor is often used as part of a larger BPM solution, providing BAM capabilities for business processes executing in various runtime environments. WebSphere Business Monitor V6.1 adds new capabilities to help keep monitor models synchronized with the applications being monitored. For example, you might need to change a business process after you invested in defining how that process should be monitored.

With the new capabilities of the WebSphere Business Monitor development toolkit, you can automatically refactor and synchronize your monitor model with the application being monitored. Table 2-4 shows the business process infrastructures that are supported by WebSphere Business Monitor V6.1.

Table 2-4 Supported business process infrastructures

Infrastructure	Description
WebSphere Process Server	When integrated with WebSphere Integration Developer and WebSphere Process Server, WebSphere Business Monitor can track the activity of the full range of components running in WebSphere Process Server. Prior releases only monitor a subset of the various types of components supported by WebSphere Process Server. WebSphere Business Monitor V6.1 supports the full set of Service Component Architecture (SCA) component, including new support for business state machines and business rules. Support for monitoring of human tasks has been improved, including a new dashboard view specifically devoted to status and actions that you can take on human tasks running in WebSphere Process Server.
WebSphere Business Services Fabric	WebSphere Business Services Fabric exploits the underlying WebSphere Process Server capability for sending business events to WebSphere Business Monitor. You can use WebSphere Business Services Fabric to assemble and manage composite business applications in a dynamic fashion and use the same BAM capability as with other WebSphere Process Server applications.
FileNet P8	The FileNet P8 V4.0 documentation explains how to enable FileNet P8 processes for event emission and business monitoring. FileNet P8 includes a monitor model that can be used to calculate metrics and track the flow of P8 processes. This monitor model can be imported into the WebSphere Business Monitor development toolkit for customization as needed.
WebSphere MQ Workflow	WebSphere Business Monitor now provides first-class support for monitoring WebSphere MQ Workflow processes. Migration tools are also provided for the benefit of WebSphere Business Monitor V4.2.4 customers. Because WebSphere Business Monitor V6.1 provides support for monitoring WebSphere MQ Workflow, such customers can upgrade and experience all the benefits of this new release. Customers can migrate both their prior V4.2.4 monitor models and the data collected by such models into the V6.1 environment.

Monitor business applications

In an SOA environment, often many of the services being aggregated into a composite business solution are hosted in remote, non-WebSphere-based infrastructures. In these cases, it is often desirable to communicate with such environments with WebSphere Adapters, which offer a wide range of connectors to many kinds of application hosting environments. It is usually best to expose adapter-based interactions as services exposed in a WebSphere Enterprise Service Bus. We present examples of such an approach in the following sections.

WebSphere Enterprise Service Bus and WebSphere Adapters

WebSphere Business Monitor provides illustrations and starting point examples for monitoring the activity in business applications, such as SAP®, by integrating with WebSphere Adapters and either WebSphere Enterprise Service Bus or WebSphere Process Server. WebSphere Enterprise Service Bus encourages exposing services via the SCA programming model. Support is built into WebSphere Enterprise Service Bus to enable event emission containing input parameters to SCA services and their return values.

WebSphere Business Monitor V6.1 also provides new support for invoking services exposed by the SCA programming model when business situations have been detected.

WebSphere Message Broker and Adapters

SupportPac™ IA9V provides a reusable Message Broker subflow for sending business events to WebSphere Business Monitor.

To learn more about IA9V, refer to the IBM WebSphere Message Broker - Common Base Event Publisher Web page at the following address:

http://www-1.ibm.com/support/docview.wss?rs=977&context=SSKMAB&context=SS7J6S&context=SSKM8N&context=SSFKSJ&context=SSFKUX&context=SSWHKB&context=SSVLA5&dc=D410&q1=SupportPac+IA9V&uid=swg24016414&loc=en_US&cs=utf-8&lang=en

WebSphere DataPower Integration Appliance XI50

The DataPower® XI50 appliance provides integration capability that can emit a business event for WebSphere Business Monitor to consume.

Monitor other business activity

In some cases, business activity is not yet naturally connected with a process server or an enterprise service bus (ESB). WebSphere Business Monitor V6.1 offers the following additional connectivity options:

- You can use WebSphere MQ as an on-ramp to send events to the WebSphere Business Monitor event infrastructure. WebSphere MQ has been used pervasively for years for various solutions, including application integration, business to business, and SOA connectivity. Many enterprises have existing WebSphere MQ environments and skills. With WebSphere Business Monitor V6.1, a business event can now be placed onto a WebSphere MQ queue and processed by WebSphere Business Monitor.
- Another means of delivering events to WebSphere Business Monitor is with environments that use WS-Notification. A Web service can publish a business event as a notification message, and the application server can stream the message to the input JMS queue for CEI. Refer to the WebSphere Application Server Information Center at the following address for information about using WS-Notification with CEI:

  `http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp`
- The IBM Tivoli portfolio of monitoring solutions has a broad reach and visibility to the various layers of IT architectures. For example, IBM Tivoli Composite Application Management (ITCAM) for SOA can identify the source of bottlenecks or failures and pinpoint services that use the most time resources. This type of monitoring is complementary to WebSphere Business Monitor, which targets business-level performance and is designed to run alongside the business interfaces used by business users, such as SAP, WebSphere Portal applications, and so on.

  ITCAM and WebSphere Business Monitor have introduced a point of integration between the products, providing you with new advantages. For example, if an IT resource or service problem is detected by ITCAM, an event can be sent from ITCAM to WebSphere Business Monitor and processed. Consider a case where the event is found to be relevant to the performance of the business. For example, performance degradation is causing insurance claims processing to suffer and you must add representatives to the call center. In this case, then WebSphere Business Monitor can alert the appropriate business user of the problem.
- For other cases, a simple Java event emitter interface is provided. As described earlier, WebSphere Business Monitor V6.1 can use XSD to describe the event schema and XML documents to be used in the payload of events. These practices are also supported for custom Java event emitters.

Pull and augment business data with user-defined functions

In prior releases, the only way to monitor an environment was to have that environment push events to WebSphere Business Monitor. A major capability in WebSphere Business Monitor V6.1 is the ability to pull data by using user-defined XPath functions within map expressions, triggers, or other places where XPath is used within the WebSphere Business Monitor programming model.

A customer can author a Java method that uses any API it desires (JDBC™, JCA, SCA, REST, and so on) to communicate with a system being monitored. Then the customer can annotate that method with Java 5 annotations and describe how it can be invoked as a new type of function available for use in XPath expressions, including being offered in content-assist when authoring such user-defined functions.

This new capability has the following uses:

- Augmenting data that is currently being delivered from events

  For example, an event might include an employee's serial number, but not include sensitive information about that employee's hourly rate. With a user-defined XPath function, the cost of working on a given activity can be computed and stored to a metric.

- As part of a recurring time-based trigger, such as polling a system to see if a particular business situation has occurred

  Such user-defined XPath functions can be used either from within (instance-level) monitoring contexts, or from within (aggregate-level) KPI contexts.

2.3.3 Dashboards

WebSphere Business Monitor V6.1 offers two dashboard deployment choices: a portal-based dashboard and a Web dashboard (without requiring portal). Both dashboards provide identical monitoring capabilities as explained in the following sections.

Archived

Getting Started guide

In WebSphere Business Monitor V6.1, the user experience has been revamped with an improved look and feel for higher responsiveness and ease of personalization. A new Getting Started guide is installed in the dashboard environment. This collection of videos and instructional guidance is a convenient way to learn what you can accomplish by using dashboards. Figure 2-18 shows the Getting Started main page with content areas to explore.

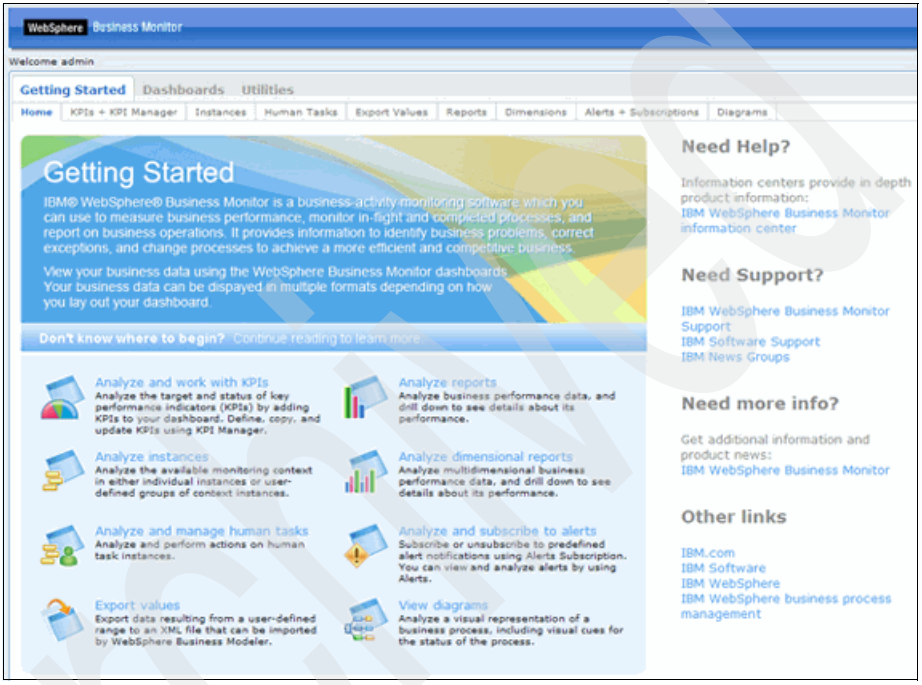


Figure 2-18 New Getting Started guide

The Getting Started guide includes learning modules for eight of the most common tasks for users with line of business responsibilities. It also demonstrates how power users can customize the BAM experience and create new KPIs. Users can navigate the Getting Started guide by using the tabs at the top of the page or clicking the icons in the content areas.

Figure 2-19 shows an example of learning about the gauge visualization for KPIs. The display is active, showing a movie of the dashboard in action. The movie cycles through the aspects of KPI visualization, including display modes, getting access to greater details, and personalization.

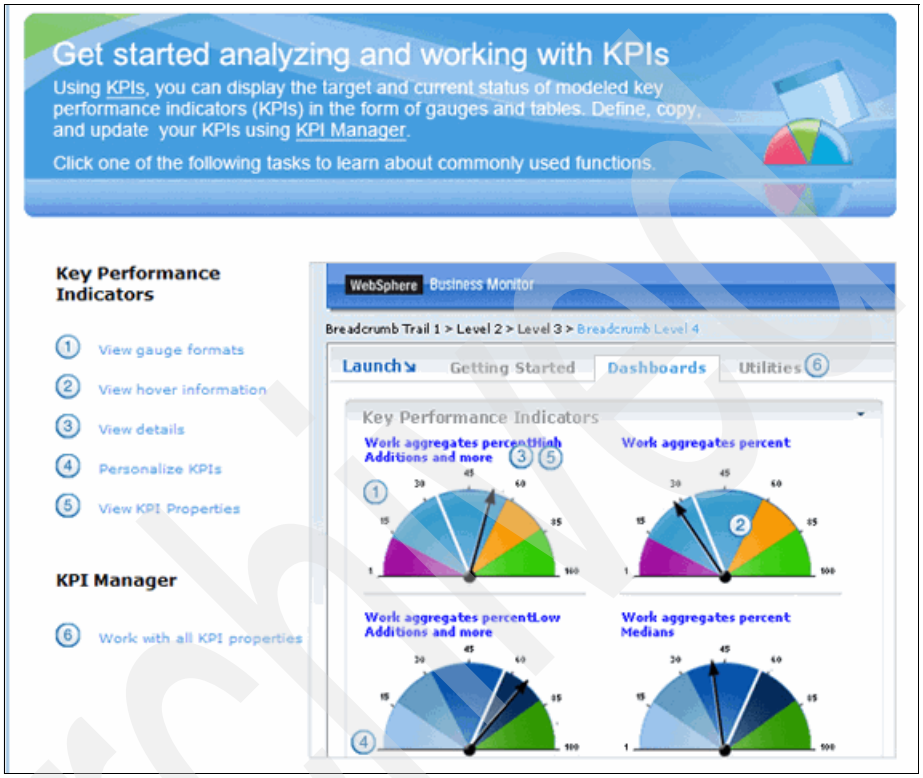


Figure 2-19 KPIs - Getting Started

The Getting Started guide also gives users a way to learn progressively. As they begin to use KPIs and become comfortable with basic features, they can refer back to Getting Started to learn more.

Revamped visualizations with Web 2.0 foundation

In WebSphere Business Monitor V6.1, the individual visual components in dashboards were redesigned with a new Web 2.0 foundation. As the WebSphere Business Monitor server processes events, dashboards are dynamically updated through a set of REST services. The dashboard visual items are implemented with Dojo and Ajax technology to provide a more visually appealing and responsive experience. For example, the display of KPIs and metrics on dashboard pages is automatically updated as events are processed. The

minimum page refresh leads to a richer user experience with the new Web 2.0 dashboards.

A dashboard page is composed from a palette of dashboard views. With WebSphere Business Monitor V6.1, creating a dashboard is a simple point-and-click or drag-and-drop operation. For example, you can create a Web dashboard by selecting New from the Web dashboard manager. You then see an empty dashboard layout. From here, you can click add-to-dashboard in the dashboard layout and choose a dashboard view from the drop-down list. Alternatively, you can drag a dashboard view from the palette to the desired location in the layout.

After a dashboard view is added to a dashboard page, it can be personalized. When personalizing a dashboard view, you bind the WebSphere Business Monitor data of interest and configure the design and functions of the dashboard view.

A dashboard page can be created for personal use, ensuring that the page is only visible to the creator. A dashboard can also be shared with a group of users. The Web dashboard manager provides a simple user interface to make a dashboard page shared or personal. In the portal environment, you can use the portal administration utilities to make a page shared.

Archived

Figure 2-20 shows an example dashboard that combines human task instances with a visual diagram. The Human Tasks view has an Actions drop-down that lets you transfer work items, or suspend and resume work items. The diagram is updated dynamically to display metrics and KPIs and change the appearance of shapes as business events are processed. Users can drag an item from the palette on the right to their page to construct a new dashboard.

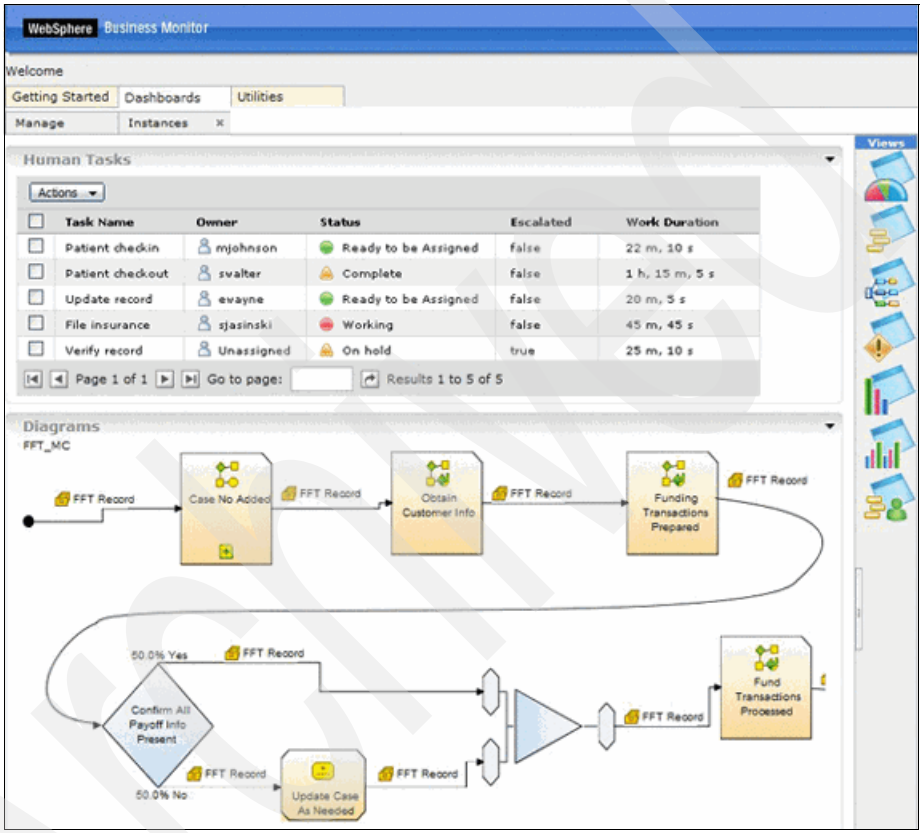


Figure 2-20 Operations dashboard layout

Arrived

Adding and changing KPIs

A key improvement in WebSphere Business Monitor V6.1 empowers business users in day-to-day operations with much less IT involvement. Business managers and executives are responsible for defining their business objectives, which can be measured in real time by using KPIs in WebSphere Business Monitor. WebSphere Business Monitor V6.1 introduces *KPI Manager*, a new feature under the Utilities menu, as shown in Figure 2-21.

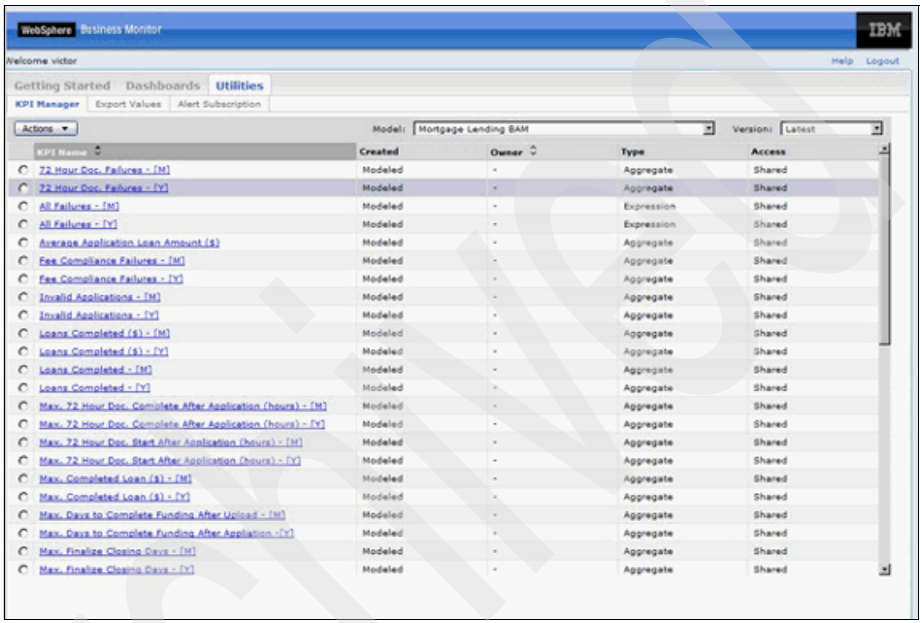


Figure 2-21 KPI Manager

By using the KPI Manager, an authorized user can create and change KPIs in a production system without IT involvement. Prior to WebSphere Business Monitor V6.1, KPIs were created and changed by using the Monitor Model editor. Then the corresponding WebSphere Business Monitor applications were deployed or redeployed by the IT department in an organization. Allowing a business user to define a KPI provides many benefits, such as a shorter development life cycle. In addition, information about business objectives is kept within the realm of business users.

From the KPI Manager, you can see that all the KPIs defined by developers are part of the original monitor models, called *modeled KPIs*. You can change the display name, target, and range, and change the color and icons associated with each range, for the modeled KPIs.

You can also define new KPIs, known as *dashboard KPIs*. No deployment is involved with the dashboard KPIs. After a dashboard KPI is defined, it becomes active and can be used in a dashboard immediately. A dashboard KPI can be created for personal use and is not visible to other people in your organization. A KPI can also be made sharable if the owner of the KPI is a KPI administrator. When a new monitor model is deployed, the dashboard KPIs is automatically merged with the new version of the monitor models. WebSphere Business Monitor V6.1 also provides a utility to transfer the dashboard KPIs to another WebSphere Business Monitor database.

An example of a business user extending monitoring capability through the KPI Manager involves adding new KPIs to the Global Human Task monitor model. This monitor model is part of the WebSphere Business Monitor server in V6.1. It contains over 20 common business metrics for all human tasks. Using these metrics, a business user can add specific KPIs for a particular human task.

Alerts and actions

Alerts are situations that require attention from a business user or notification to an external system. A WebSphere Business Monitor IT administrator can set up templates to define how each type of alert can be handled, such as dashboard, e-mail, cell, pager, SCA service, and Web service. A business user can subscribe to alerts to be delivered by dashboard, e-mail, cell, and pager.

Prior to WebSphere Business Monitor V6.1, subscribing to alerts can only be performed by WebSphere Application Server administrators on behalf of the business user. A business user in WebSphere Business Monitor V6.1 can now subscribe to the business alerts and choose one or more delivery methods.

As shown in Figure 2-22 on page 51, when alerts are delivered to business users by a dashboard, they can see the alerts from an Alert view. A business user can mark each alert as read or unread and can forward or remove an alert.

Role-based dashboard access

A WebSphere Business Monitor IT administrator can control access to dashboards based on the business user's role. Access can be specified on a model level. For example, a user can be a KPI administrator for one monitor model but have no access to another model. When binding a monitor model to a dashboard view, only the authorized monitor models are made available to the business users. Operations, such as modifying a shared KPI, are disabled for unauthorized business users.

Advanced integration with WebSphere Portal

Each WebSphere Business Monitor dashboard view can also be deployed as a portlet to WebSphere Portal Server. All the WebSphere Business Monitor portlets are in the portlet palette. With the significantly improved page creation

interface in WebSphere Portal Server V6.0.1, a business user can create a Portal dashboard page by dragging the WebSphere Business Monitor portlet from the palette to the desired area of a page. Each portlet can then be personalized, which involves binding the data to the dashboard view and configuring the visualization of the dashboard view. In the Portal environment, you can limit users to view access, and not allow them to change or personalize a dashboard view.

A Portal dashboard may contain visualizations (portlets) provided by WebSphere Portal Server, as well as custom portlets. When WebSphere Portal Server is used, the dashboard utilities, such as the KPI Manager and Export Values, are also installed as portlets.

Figure 2-22 shows a Portal dashboard page with the WebSphere Business Monitor Alerts view and KPIs view. It also has a World Clock portlet and a Person Finder portlet. For customers who need to create custom monitor visualizations, or to combine visualization from other sources, WebSphere Portal Server provides a platform for rich dashboard customization.

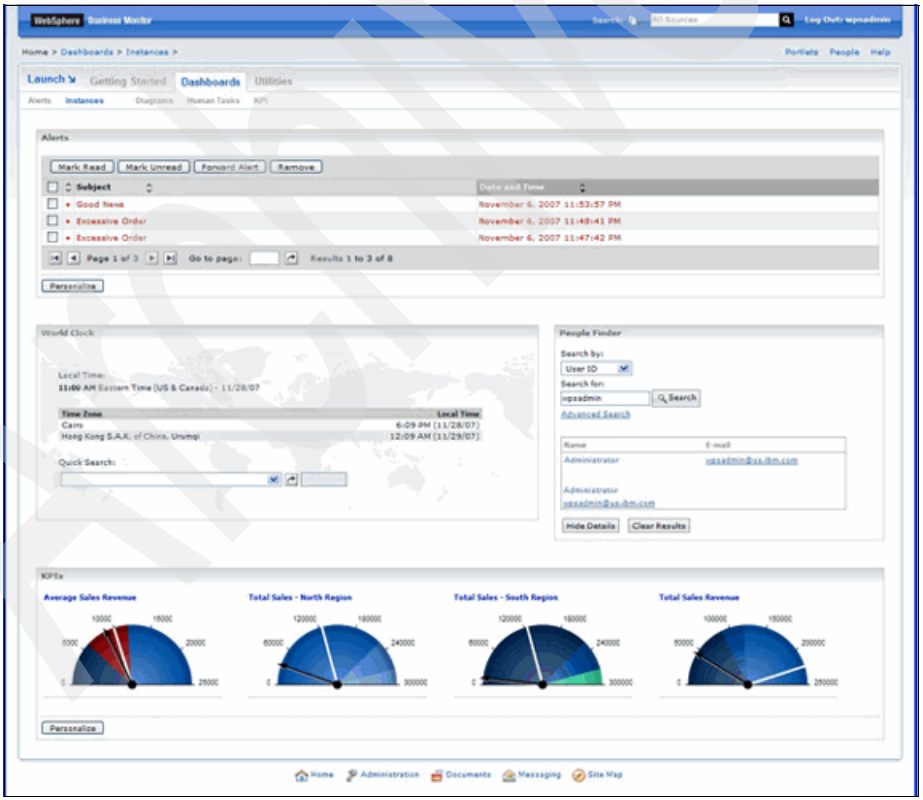


Figure 2-22 Portal-based dashboard

2.3.4 Installation and administration

WebSphere Business Monitor V6.1 installation is consistent with other related products. Administrative tasks have also been streamlined and automated.

Consistency

The WebSphere Business Monitor V6.1 installer focuses on consistency and reliability, giving you a good installation experience.

Users of WebSphere Application Server V6.1 and related products will notice that the usage, navigation, and panel design in WebSphere Business Monitor V6.1 is consistent with WebSphere Application Server V6.1. From the welcome page to the First Steps window, users are not required to learn a new approach for installation. Likewise, maintenance on the WebSphere Business Monitor V6.1 server is performed by using the same Update Installer that is used with WebSphere Application Server V6.1.

WebSphere Business Monitor V6.1 supports consistent WebSphere profile management for WebSphere Business Monitor server configurations. Tasks, such as creating or augmenting a deployment manager profile or a stand-alone application server profile, are performed by using the same WebSphere profile management tool that is used with WebSphere Application Server V6.1.

On the authoring side, the WebSphere Business Monitor development toolkit has been designed for consistency with other development tools, such as Rational Application Developer V7.0 and WebSphere Integration Developer V6.1. Both of those tools have adopted a common installation approach with the IBM Installation Manager. The result is a common design, function, and approach for problem determination and uninstallation.

Range of installation scenarios

A highlight of WebSphere Business Monitor V6.1 is a reduction in the footprint of the required software. In version 6.0.2, a WebSphere Business Monitor server installation always required DB2, DB2 Alphablox, DB2 Cube Views™, WebSphere Process Server, WebSphere Portal Server, and an LDAP server. Now, the only mandatory prerequisite is WebSphere Application Server V6.1. You can still choose to install DB2, DB2 AlphaBlox® and WebSphere Portal Server if your project requires the advanced capability. DB2 Cube Views is no longer used by WebSphere Business Monitor and is not part of the installation experience.

As with V6.0.2, the WebSphere Business Monitor V6.1 server installation still has the choice of a basic or advanced installation path. With the basic installation path, the installation program uses common default options and installs the prerequisite software on behalf of the user. This path is a quick and easy way to

install a single WebSphere Business Monitor server topology. The advanced installation path is used for the typical production-style topologies where WebSphere Business Monitor components are distributed to different servers.

Support for broader operating environments

WebSphere Business Monitor V6.1 provides expanded support for operating systems and database platforms. Additional key operating system support includes Red Hat Enterprise Linux®, SUSE® Linux Enterprise Server, and HP-UX Itanium™.

Key WebSphere Business Monitor server supported database platforms are DB2 for z/OS®, Oracle 10g Standard and Enterprise, Derby Embedded (for non-production usage only), and version 9.1 of DB2 Universal Database™.

First steps

A popular new feature to be used immediately after installation is the First steps console. As shown in Figure 2-23, the First steps console includes a list of actions and resources to help accelerate your BAM project.

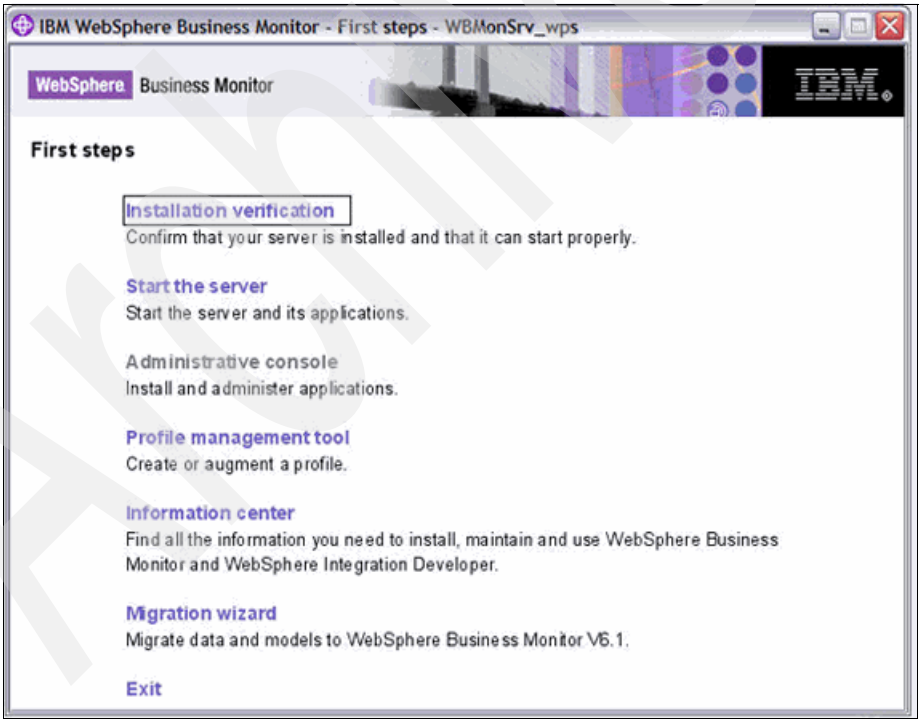


Figure 2-23 WebSphere Business Monitor First steps

The following First steps items are new in WebSphere Business Monitor V6.1:

- The *installation verification utility* can help identify any possible installation problems before continuing with the project. The utility deploys a monitor model and processes events through the model to exercise much of the WebSphere Business Monitor server function. If this verification passes successfully, you can then proceed with installing monitor models and configuring the WebSphere Business Monitor server.
- The *Migration Wizard* can be used to migrate data and models from WebSphere Business Integration Monitor V4.2.4 and WebSphere Business Monitor V6.0.2. The wizard walks an administrator through the steps of migrating and verifying the updated resources on WebSphere Business Monitor V6.1.

2.4 Summary

In this chapter, we introduced the architecture and basic concepts of WebSphere Business Monitor V6.1. We also discussed the new features in WebSphere Business Monitor V6.1.

For additional articles that discuss the new capabilities in depth, refer to the Architecture Technical library view in IBM developerWorks® at the following address:

http://www.ibm.com/developerworks/views/architecture/libraryview.jsp?search_by=put+new+%20capabilities+of+bam+to+work

Archive

3

Using WebSphere Business Monitor

In this chapter, we provide an overview of how to install and configure WebSphere Business Monitor. We begin by discussing a typical installation deployment topology for WebSphere Business Monitor. We explain how to install the toolkit and WebSphere Business Monitor. We also explain how to deploy and administer monitor models and how to monitor with Web-based dashboards.

© Copyright IBM Corp. 2008. All rights reserved.

3.1 Installation deployment topology for WebSphere Business Monitor

In this section, we describe possible installation deployments of WebSphere Business Monitor. The following topologies illustrate common installations:

- Development topology
- Single server topology
- Distributed topologies

3.1.1 Development topology

The development topology describes the installation for the WebSphere Business Monitor development toolkit. Two components in the WebSphere Business Monitor development toolkit can be installed on the same system: the WebSphere Business Monitor development environment and test environment. You must install the Monitor development environment, but the Monitor test environment is optional.

You can use either Rational Application Developer or WebSphere Integration Developer as your development platform. In both platforms, you have the option to install the Monitor test environment. You can only install the Monitor test environment if you installed the test environment for the development platform. Although the test server platforms differ, you can install the Monitor test environment. When the Monitor test environment is installed, the WebSphere Business Monitor server is installed into a test server profile, and the WebSphere Business Monitor database is created.

Archived

Figure 3-1 illustrates the development topology.

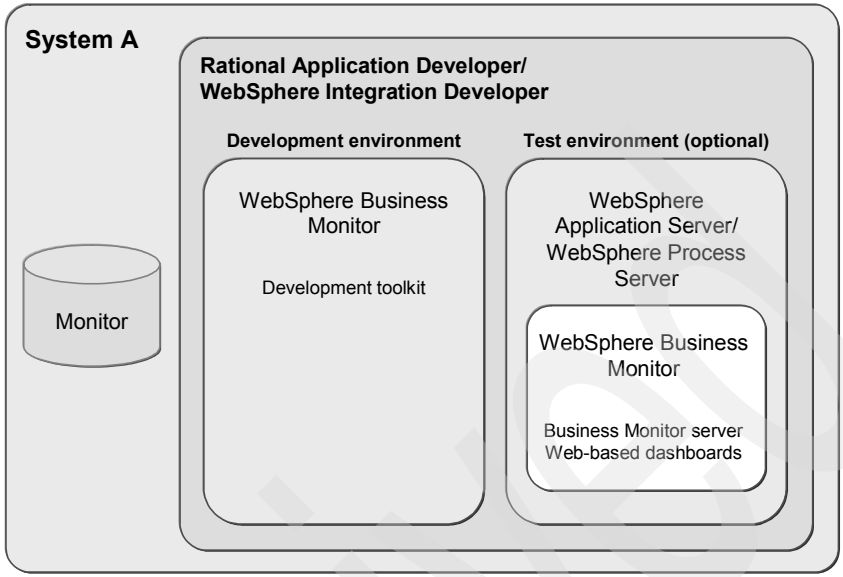


Figure 3-1 The development topology

3.1.2 Single server topology

By using the single server topology, all supporting products and all WebSphere Business Monitor components are installed on the same physical server. Installing WebSphere Business Monitor on a single server is ideal for development test environments, proof-of-concept environments, and simple deployments that do not require failover and high availability capabilities.

You can use the WebSphere Business Monitor installation program to install all WebSphere Business Monitor components and WebSphere Application Server. To view the monitored data, you can use either the Web-based dashboards or the portlet-based dashboards. For the Web-based dashboards, WebSphere Application Server is required. For this topology, you can install a single instance of WebSphere Business Monitor server and the Web-based dashboards on a single instance of WebSphere Application Server.

If you install the portlet-based dashboards, the installation program can install WebSphere Portal. Similarly, if you install the Alphablox features for your dashboards, the installation program can install DB2 Alphablox. You can install the Alphablox features for either Web-based dashboards or portlet-based dashboards.

In most cases, to install a WebSphere Business Monitor on a single server, you can use the basic option during the installation process. Using the basic installation option installs WebSphere Application Server, WebSphere Portal, and DB2 Alphablox depending on your choices during the installation. To use existing installations of the prerequisite products when installing WebSphere Business Monitor, you must use the advanced installation option.

If you are using a database software other than Derby Embedded, you must install the database software before running the installation program for WebSphere Business Monitor. Derby Embedded is ideal for development and proof-of-concept environments.

Figure 3-2 illustrates the distribution on one physical server.

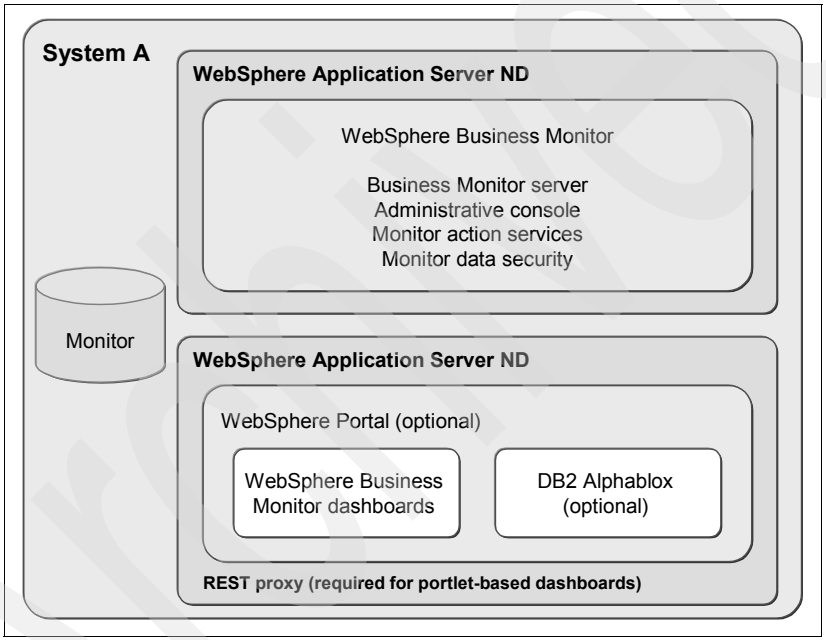


Figure 3-2 The single server topology

The following required products are installed on System A:

- WebSphere Application Server ND
- One of the following supported databases:
  - IBM DB2 Universal Database Enterprise Server Edition
  - Derby Embedded
  - Oracle Server

If you are installing the portlet-based dashboards, you must also install WebSphere Portal and the REST proxy on System A.

If you are installing the Alphablox Dashboards, you must also install DB2 Alphablox on System A.

The following WebSphere Business Monitor components are installed on System A:

- WebSphere Business Monitor server
- Data movement services
- WebSphere Business Monitor data security
- WebSphere Business Monitor action services
- Dashboards
- WebSphere Business Monitor database

3.1.3 Distributed topology

To improve performance, you can install the WebSphere Business Monitor components on different servers. You can use a two-server distribution or a three-server distribution.

You can use the WebSphere Business Monitor installation program to install all WebSphere Business Monitor components and WebSphere Application Server. To view the monitored data, you can use either the Web-based dashboards or the portlet-based dashboards. For the Web-based dashboards, WebSphere Application Server is required on the server where the Web-based dashboards will be installed. If you are installing the portlet-based dashboards in this topology, you must install WebSphere Portal prior to running the WebSphere Business Monitor installation program. Similarly, if you are installing the Alphablox features for your dashboards, you must install DB2 Alphablox prior to running the WebSphere Business Monitor installation program.

Archived

Installing WebSphere Business Monitor on two servers

Figure 3-3 illustrates a two-server distributed topology.

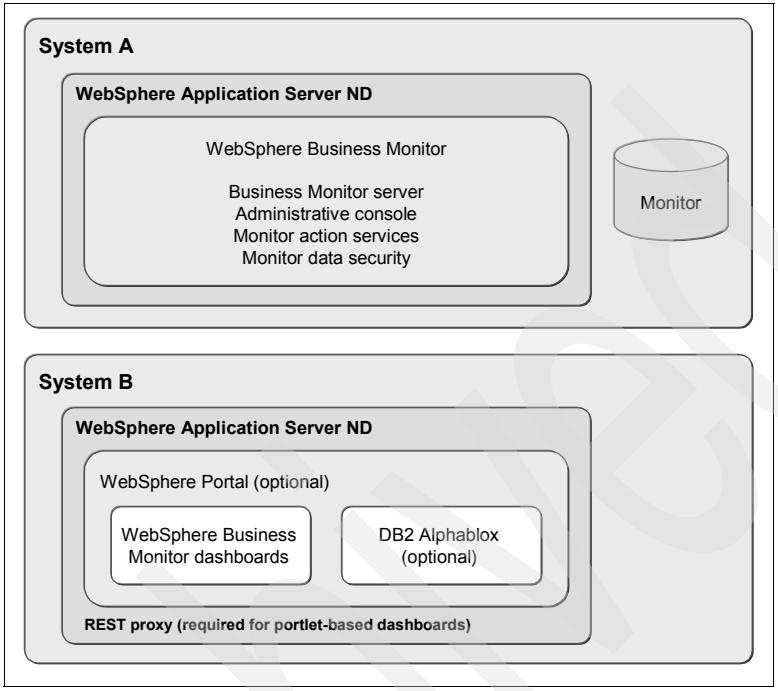


Figure 3-3 The two-server distributed topology

The following required products are installed on System A:

- WebSphere Application Server ND
- One of the following supported databases:
  - IBM DB2 Universal Database Enterprise Server Edition
  - Derby Embedded
  - Oracle Server

If you are using a database software other than Derby Embedded, you must install the database software before running the installation program for WebSphere Business Monitor. Derby Embedded is ideal for development and proof-of-concept environments.

The following WebSphere Business Monitor components are installed on System A:

- WebSphere Business Monitor server
- Data movement services
- Monitor data security

Archived

- Monitor action services
- Monitor database

WebSphere Application Server ND is a required product and is installed on System B. If you are installing the portlet-based dashboards, you must also install WebSphere Portal and the REST proxy on System B.

If you are installing the Alphablox features for the dashboards, you must also install DB2 Alphablox on System B. You can install the Alphablox features for either Web-based dashboards or portlet-based dashboards. If you choose to install both Web-based dashboards and portlet-based dashboards, you can install DB2 Alphablox on each server, but you must configure multiple instances of DB2 Alphablox to use the same database.

The Dashboards component of WebSphere Business Monitor is installed on System B.

Installing WebSphere Business Monitor on three servers

Figure 3-4 illustrates a three-server distributed topology.

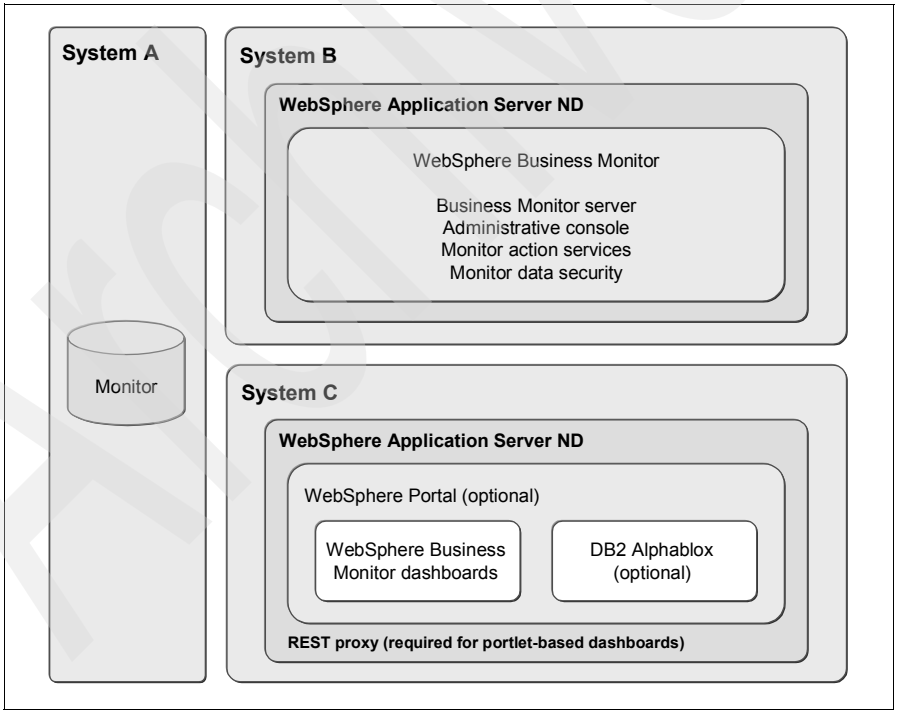


Figure 3-4 The three-server distributed topology

System A requires one of the following supported databases

- IBM DB2 Universal Database Enterprise Server Edition
- Derby Embedded
- Oracle Server

If you are using database software other than Derby Embedded, you must install the database software before running the installation program for WebSphere Business Monitor. Derby Embedded is ideal for development and proof-of-concept environments.

The WebSphere Business Monitor database component of WebSphere Business Monitor is installed on System A.

WebSphere Application Server is a required product and is installed on System B. The following WebSphere Business Monitor components are installed on System B:

- WebSphere Business Monitor server
- Data movement services
- Monitor data security
- Monitor action services

WebSphere Application Server is a required product and is installed on System C. If you are installing the portlet-based dashboards, you must also install WebSphere Portal and the REST proxy on System C.

If you are installing the Alphablox features for the dashboards, you must also install DB2 Alphablox on System C. You can install the Alphablox features for either Web-based dashboards or portlet-based dashboards. If you choose to install both Web-based dashboards and portlet-based dashboards, you can install DB2 Alphablox on each server. However, you must configure multiple instances of DB2 Alphablox to use the same database.

The Dashboards component of WebSphere Business Monitor is installed on System C.

3.2 Installing the toolkit

For a successful installation of WebSphere Business Monitor development toolkit V6.1.1, you must have the required system and software and properly following the installation steps.

WebSphere Business Monitor development toolkit provides wizards, libraries, and a test environment to help you create and test monitor models. WebSphere

Business Monitor development toolkit can be installed in an existing Rational Application Developer or WebSphere Integration Developer environment.

The following two components comprise the WebSphere Business Monitor development toolkit:

- Monitor development environment
- Monitor test environment

The WebSphere Business Monitor development environment includes the Monitor Model editor and Monitor Application Descriptor models. The Monitor test environment includes the WebSphere Business Monitor server, which includes the WebSphere Business Monitor action services and Web-based dashboards.

With the Monitor test environment, you can deploy your monitor model to a WebSphere Business Monitor server, configure WebSphere Business Monitor action services, and view the data in the Web-based dashboards with DB2 Alphablox. By using DB2 Alphablox, you can add Reports, Dimensions, and Organizations to your dashboards.

The Monitor test environment also supports the following tasks:

- Promotes fast and iterative development (using the Republish support)
- Allows you to fill in test events and emit them
- Provides support for launching the Common Base Event browser within your Rational Application Developer or WebSphere Integration Developer environment

To install the Monitor test environment, you must have installed one of the following options:

- WebSphere Process Server test environment in WebSphere Integration Developer
- WebSphere Application Server test environment in Rational Application Developer

In this section, we explain how to install only the toolkit into WebSphere Integration Developer.

More information: For the latest information about installing the toolkit into WebSphere Integration Developer and Rational Application Developer, see the WebSphere Business Monitor Information Center at the following address:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp`

Archive

3.2.1 System requirements

You can install the WebSphere Business Monitor development toolkit in an existing WebSphere Integration Developer environment. The WebSphere Business Monitor development toolkit is available for Microsoft® Windows® only.

Before you begin the installation, you must ensure the installation of the following items:

- To install the toolkit, you must have installed WebSphere Integration Developer V6.1.
- To install the Monitor test environment, an optional component of the toolkit, you must have installed the WebSphere Process Server test environment for WebSphere Integration Developer V6.1.

3.2.2 Installing the toolkit into WebSphere Integration Developer

To install the toolkit into WebSphere Integration Developer:

1. Run the Launchpad by starting the batch file launchpad.bat.
2. On the page that opens (Figure 3-5), click the **Install IBM WebSphere Business Monitor Development Toolkit V6.1** link to begin the installation.

Figure 3-5 Install IBM WebSphere Business Monitor Development Toolkit V6.1

Archived

3. In the Install Packages panel (Figure 3-6), select the **IBM WebSphere Business Monitor Development Toolkit** and **Version 6.1.1.0** check boxes, and click **Next**.

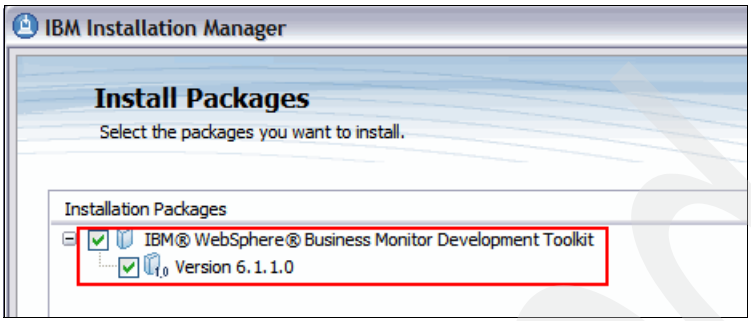


Figure 3-6 Install Packages

4. In the Licenses panel, review the terms of the license agreement, and click **I accept the terms in the license agreements**. Click **Next**.

   The installation program performs a brief check to determine which package groups are currently installed on the system.

5. In the Location panel, click **Next**.

6. In the Install Packages – Features panel (Figure 3-7 on page 66), choose from the following features to be installed:

   – Monitor Development Environment

     Installs the Monitor Development Environment. This option is selected by default and cannot be edited.

   – WebSphere Integration Developer support for monitor model generation

     Enables generation of monitor models from WebSphere Process Server and WebSphere Enterprise Service Bus applications.

   – Monitor Test Environment

     Installs the Monitor test environment into the existing testing environment.

     • WebSphere Business Monitor profile on WebSphere Process Server

       Creates the WBMonSrv_wps profile in the embedded WebSphere Process Server test server.

     • WebSphere Business Monitor profile on WebSphere ESB

       Creates the WBMonSrv_esb profile in the embedded WebSphere Enterprise Service Bus test server.

   Click **Next**.

Archive

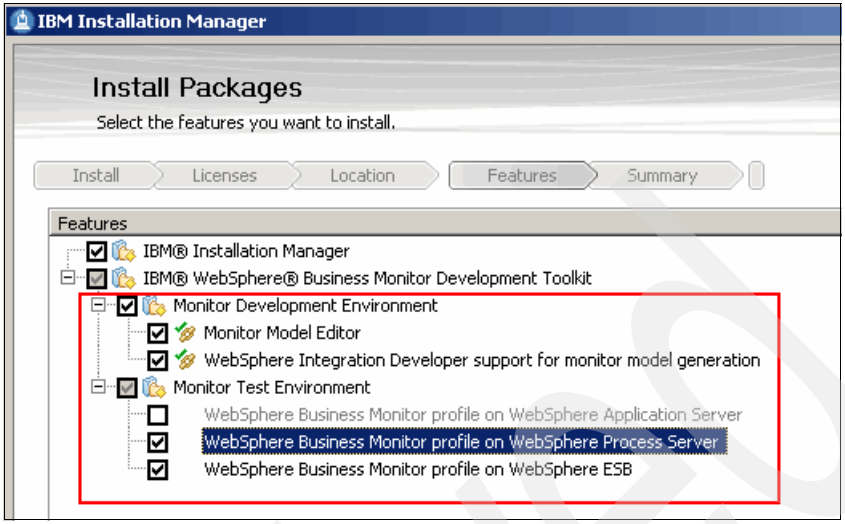


Figure 3-7 Install Packages

7. In the Summary panel, review the information about the features to be installed, the required disk space, and the installation location. Then click **Install** to begin the installation.
8. When the installation is complete, click the **View Log File** link in the last panel to verify whether there are any error messages. If you see the following error message, you can safely ignore it:

   ```
   Error expanding installation packages In root installation context:

   Component "com.ibm.was61" required by component
   "com.ibm.wbimonitor.toolkit.ite.was61" was not found

   WebSphere Application Server version 6.1.0.13 test environment is not
   installed. You must install the WebSphere Application Server version
   6.1 test environment from Rational Application Developer version 7
   and upgrade it to version 6.1.0.13.
   ```

   Click **Finish** to exit the installation.

Archived

3.2.3 Adding and removing the Monitor test environment

You can modify an existing installation of the WebSphere Business Monitor development toolkit as explained in the following steps. Use the IBM Installation Manager to add or remove the Monitor test environment.

1. Start the IBM Installation Manager. From the Windows desktop, select **Start** → **All Programs** → **IBM Installation Manager** → **IBM Installation Manager**.
2. Click **Modify Packages**.
3. Select the package group that contains IBM WebSphere Business Monitor Development Toolkit, and click **Next**.
4. In the next panel (Figure 3-8 on page 67), choose either of the following options:
   - To install the Monitor test environment, select the **Monitor Test Environment** check box.
   - To remove the Monitor test environment, clear the **Monitor Test Environment** check box.

   Click **Next**.

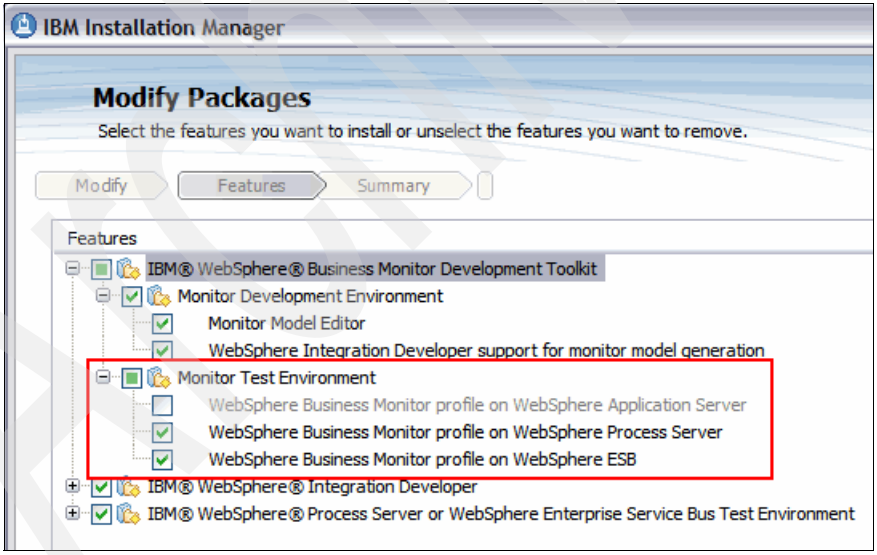


Figure 3-8 Modified Packages

5. Verify the information in the Summary panel, and click **Modify** to modify the existing installation.

Archived

3.2.4 Verifying the toolkit installation

After you install the WebSphere Business Monitor development toolkit, you must verify that the Monitor development environment and the Monitor test environment are properly installed.

Verifying the Monitor development environment

To verify the Monitor development environment:

1. Verify that no error messages are in the installation log files. There are two log files to view that are labeled with a build number (which is in the date-format) and the current time:
   - build_number.current_time.log
   - build_number.current_time.xml

   For example, on Windows operating systems, the file paths can be similar to the following examples:

   ```
   C:\Documents and Settings\All Users\Application
   Data\IBM\Installation Manager\logs\native\20071015_0947.log
   ```

   ```
   C:\Documents and Settings\All Users\Application
   Data\IBM\Installation Manager\logs\20071015_0951.xml
   ```

   Always review the file with the most recent time stamp and verify that the *WBMonSrv* profile has been created successfully.

   The following example log file is based on the WebSphere Business Monitor development toolkit installed into a WebSphere Integration Developer environment:

   ```
   INSTCONFSUCCESS: Success: Profile WBMonSrv_wps now exists. Please
   consult C:\IBM\WID61\pf\WBMonSrv_wps/logs/AboutThisProfile.txt for
   more information about this profile.
   ```

   ```
   INSTCONFSUCCESS: Profile augmentation succeeded.
   ```

   ```
   Successfully create the Monitor profile, WBMonSrv_wps, for the
   Monitor Test Environment. See the log file in the following location
   for details: "C:\Documents and Settings\All Users\Application
   Data\IBM\Installation
   Manager\logs\native"\wbmonitor\bi_v61\logs\manageprofiles
   ```

2. Verify that you can open the Business Monitor perspective:
   a. From the Windows desktop, click **Start** → **WebSphere Integration Developer**.
   b. Select **Window** → **Open Perspective** → **Other**.
   c. In the Open Perspective window, select **Business Monitor**.

Archived

Verifying the Monitor test environment

To verify the Monitor test environment:

1. Verify that one of the following folders was created under the installation directory of WebSphere Integration Developer:
   - \runtimes\base_v61\profiles\WBMonSrv
   - \pf\WBMonSrv_wps
   - \pf\WBMonSrv_esb
2. Verify that there are no error messages and review the applicable systemOut.log file under the installation directory of WebSphere Integration Developer:
   - \runtimes\base_v61\profiles\WBMonSrv\logs\server1\systemOut.log
   - \pf\WBMonSrv_wps\logs\server1\systemOut.log
   - \pf\WBMonSrv_esb\logs\server1\systemOut.log
3. Start WebSphere Integration Developer and verify that WebSphere Business Monitor Server V6.1 can be started from the Servers view:
   a. In the Business Monitor perspective (Figure 3-9), click the **Servers** tab and select **WebSphere Business Monitor server v6.1 on WebSphere Process Server**. Then click the **Start** icon (green circle with white arrow button).

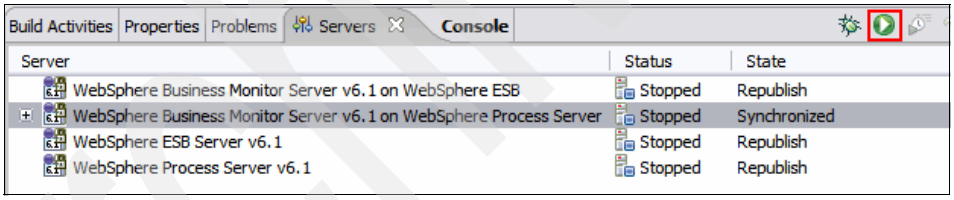


Figure 3-9 Servers

Archive

b. When the server is started, launch the WebSphere Application Server Administrative Console to verify that Monitor Models, Action Manager, Web Dashboard, Monitor Rest Services, and Alphablox Web Dashboards have been installed. From the Servers view (Figure 3-10):

   i. To launch the dashboard, right-click **WebSphere Business Monitor Server v6.1** and select **Run WBM Web Dashboard**.

   ii. To launch the Common Base Event browser, right-click the server and select **Common Base Event Browser**.

   iii. If you are using Alphablox, verify that you can open the Alphablox administrative console. Right-click **WebSphere Business Monitor Server v6.1** and select **Alphablox Admin Console**.

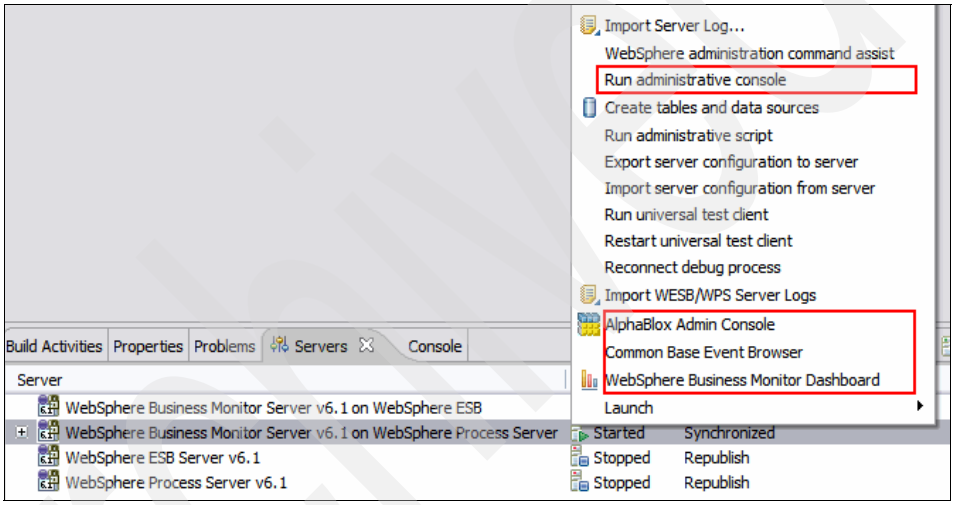


Figure 3-10 Servers

3.3 Installing WebSphere Business Monitor

In the following sections, we describe the system and software requirements necessary for installation and the steps to successfully install WebSphere Business Monitor V6.1.1.

3.3.1 WebSphere Business Monitor prerequisites

The following components are required for you to install WebSphere Business Monitor V6.1.1:

- Windows Server® 2003 Enterprise Edition with service pack 1
- Windows Server 2003 Standard Edition with service pack 1
- Windows XP Professional with service pack 2
- AIX® 5L™ 5.2 with maintenance level 5200-079
- AIX 5L 5.3 technology level 5300-05-06
- HP-UX on Itanium Hardware 11i V2 and V3
- Red Hat Enterprise Linux (RHEL) 5.0 Advanced Platform
- SUSE Enterprise Linux Server (SLES) Version 9
- SUSE Enterprise Linux Server (SLES) Version 10

Detailed system requirements: For the latest information about platform-specific disk space requirements, supported operating systems, and the operating system fixes and patches that you must install to have a compliant operating system, see WebSphere Business Monitor detailed system requirements at the following Web address:

`http://www-306.ibm.com/software/integration/wbimonitor/requirements/`

When you reach this Web page, select the link to your version of WebSphere Business Monitor.

3.3.2 Preparing to install the required software

WebSphere Business Monitor 6.1 installation includes an installation preparation wizard. This new utility helps you set up the required software for the Business Monitor installation. Instead of following the instructions to manually set up the required product images in a particular structure, with this utility, you can easily lay down all the necessary product images, in the correct way, required by the Business Monitor installation. In this section, we walk you through an example of running the installation preparation wizard. Alternatively, you can use the instructions to do this manually.

Preparing the files by using the Installation preparation utility

The WebSphere Business Monitor Installation preparation utility gathers the required and optional prerequisite software that you want to install into a single location. This location can then be used for product installation. You can use this utility with your product CDs or with an uncompressed installation image.

To move the prerequisite software to a single location by using the Installation preparation utility:

1. Start **launchpad.exe** from the installation CD or folder.
2. In the Launchpad, click **Launch the installation preparation wizard for WebSphere Business Monitor** (Figure 3-11).

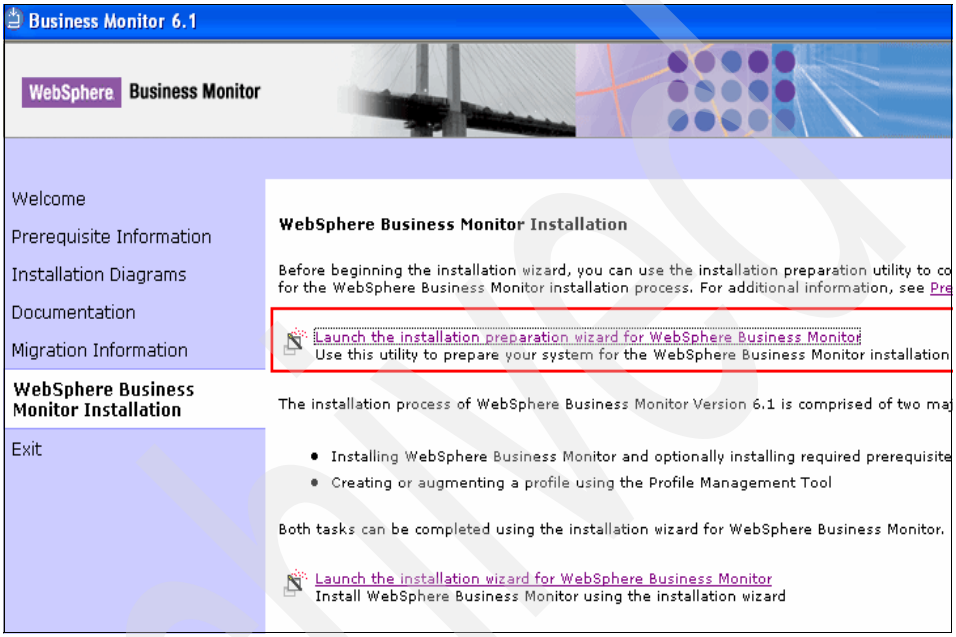


Figure 3-11 WebSphere Business Monitor Installation preparation wizard

3. On the Welcome page, click **Next**.

Archive

4. In the Product selection panel (Figure 3-12), select **WebSphere Application Server ND6.1**, **DB2 Alphablox 9.5** and **Information center**. Then click **Next**.

> **Custom selections:** In the Product selection panel, WebSphere Business Monitor 6.1 is selected by default. You can make your selections based on the type of installation that you will perform and the features you plan to install. For example, if you plan to do a basic installation without Portal Dashboard, you select WebSphere Application Server ND 6.1 and DB2 Alphabox 9.5. If you plan to do a basic installation with Portal Dashboard, then you must also select WebSphere Portal 6.0.1.1.

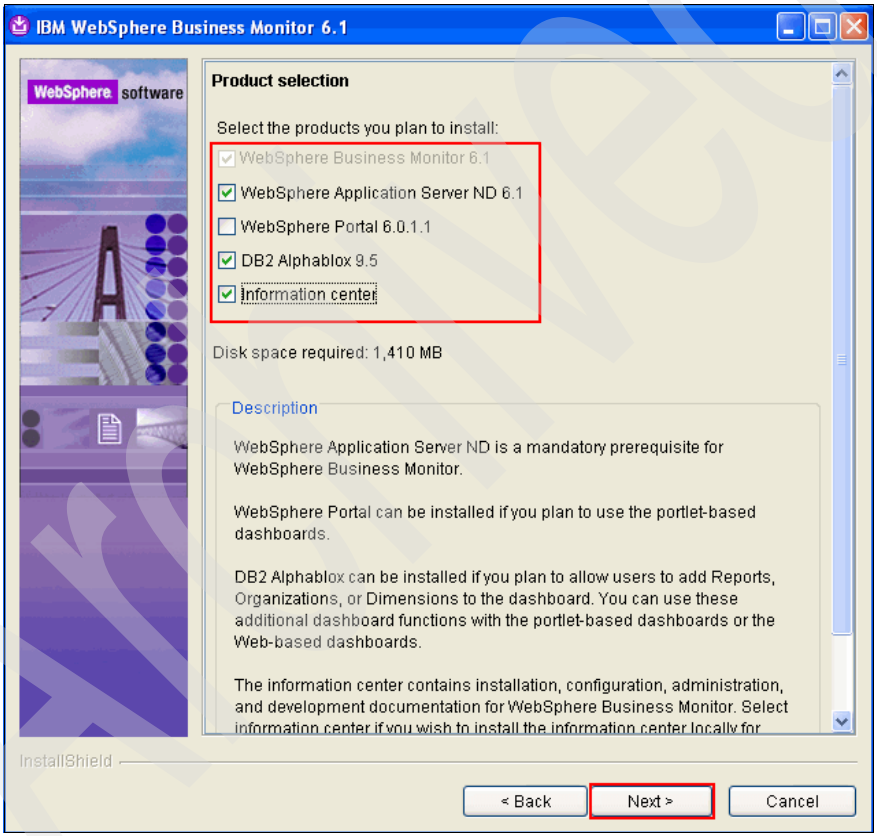


Figure 3-12 WebSphere Business Monitor 6.1 installation preparation: Product selection

5. In the Target directory panel (Figure 3-13), in the Target directory field, enter a path. In our example, we type `C:\WBM_Install_Image`. Click **Next**.

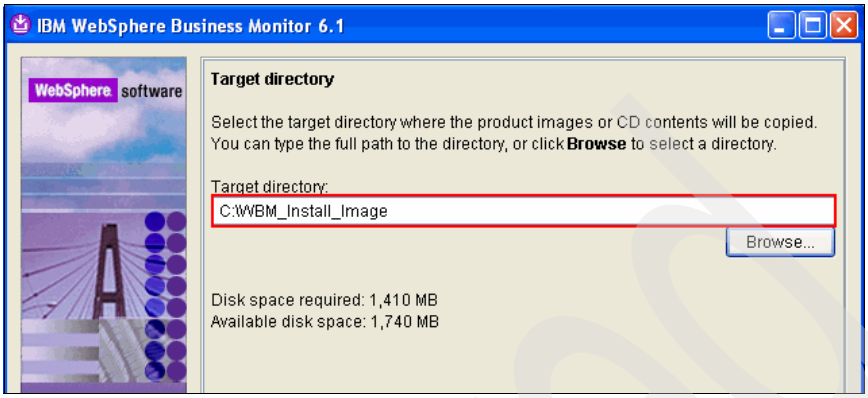


Figure 3-13 WebSphere Business Monitor 6.1 installation preparation: Target directory

6. In the Source directory panel (Figure 3-14), in the WebSphere Business Monitor Version 6.1 source directory field, enter the path of the installation image. In our example, we type `C:\WebSphere Business Monitor V6.1.1 Setup`. Click **Next**.

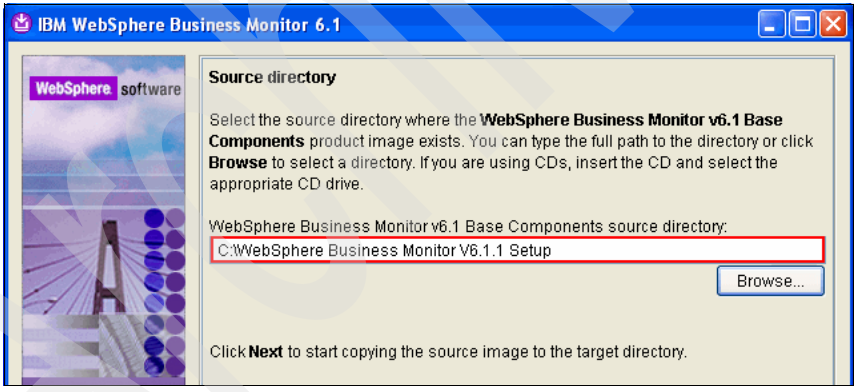


Figure 3-14 WebSphere Business Monitor 6.1 installation preparation: Source directory

While the tool is copying the product image into your target directory, you see the panel shown in Figure 3-15.

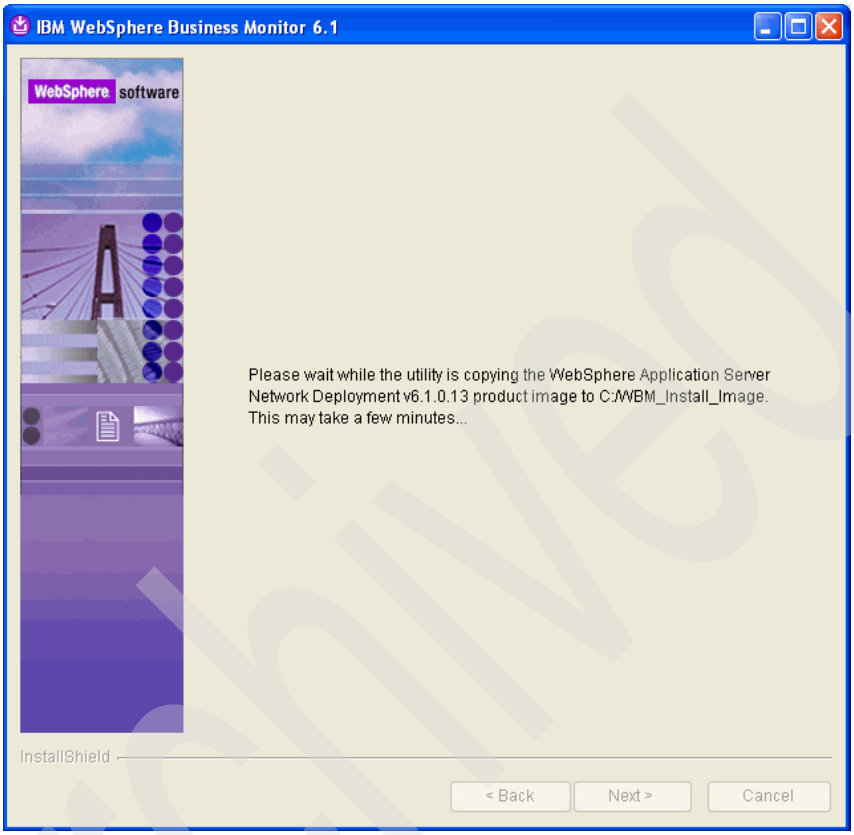


Figure 3-15 WebSphere Business Monitor 6.1 installation preparation: Copying the product image

Archive

7. In the Source directory panel (Figure 3-16), in the WebSphere Application Server Network Deployment v6.1.0.13 source directory field, type the path of the installation image. In our example, we type `C:\WebSphere Business Monitor V6.1.1 Setup`. Click **Next**.

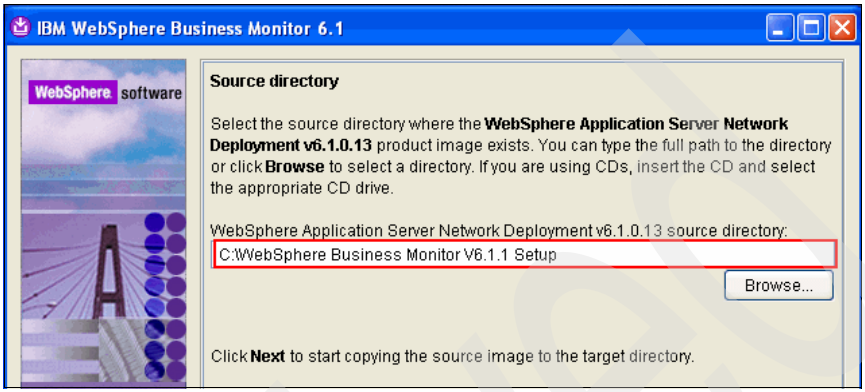


Figure 3-16 WebSphere Business Monitor 6.1 installation preparation: WebSphere Application Server Network Deployment V6.1.0.13 source directory

8. In the Source directory panel (Figure 3-17), in the Alphablox V9.5 source directory field, enter the path of the installation image. In our example, we type `C:\WebSphere Business Monitor V6.1.1 Setup`. Click **Next**.

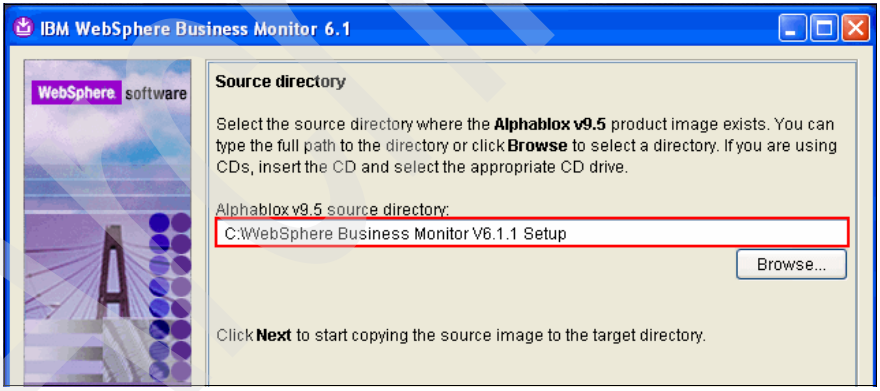


Figure 3-17 WebSphere Business Monitor 6.1 installation preparation: Alphablox V9.5 source directory

9. In the Source directory panel (Figure 3-18), in the WebSphere Business monitor v6.1 Quick Start CD source directory field, enter the path of the installation image. In our example, we type `C:\WebSphere Business Monitor V6.1.1 Setup`. Click **Next**.

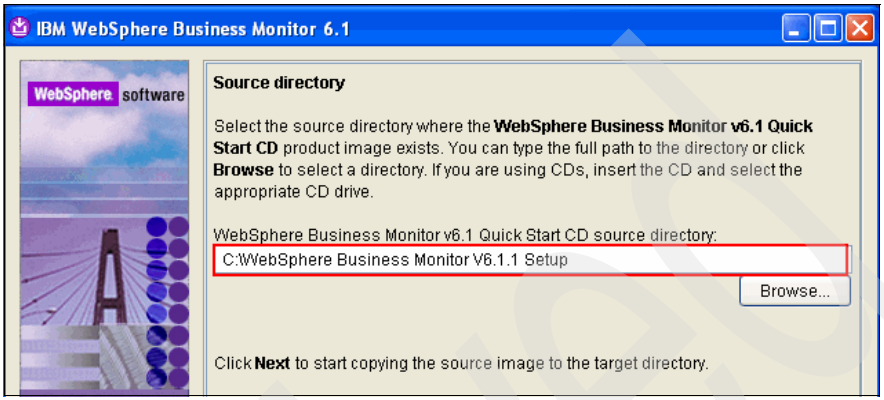


Figure 3-18 WebSphere Business Monitor 6.1 installation preparation: WebSphere Business Monitor V6.1 Quick Start CD source directory

10. The next display (Figure 3-19) shows the target directory and the list of products that the tool copied. At this point, all products that you will use in your Business Monitor installation are set up correctly. Click **Finish** to exit the tool.

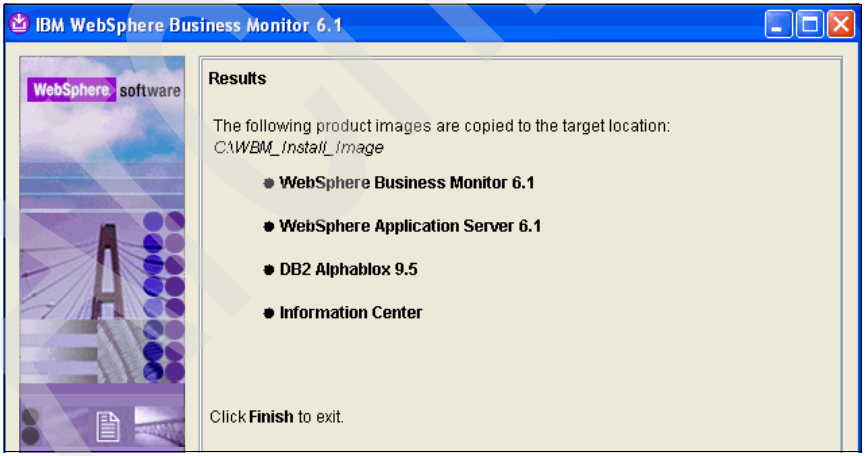


Figure 3-19 WebSphere Business Monitor 6.1 installation preparation: Result

Manually preparing files for the installation

If you do not use the Installation preparation utility to move the prerequisite software into a single location for product installation, you must manually move the files to a single directory. Copy the CD or uncompressed downloadable file contents to the directories indicated:

1. Create a Monitor_installation_image directory, for example `C:\WBM_Install_Image`.
2. Extract the files for each supporting product into the subdirectories shown in Table 3-1, in the Monitor_installation_image directory.

Table 3-1 Directory name

Product	Directory name
WebSphere Application Server	WAS
Optional: WebSphere Portal	WP Contains the following subdirectories for the different CDs and downloadable images: ► setup ► was ► base1 ► base2 ► base3
Optional: DB2 Alphabox	ABX
WebSphere Business Monitor Information Center	IFC

3. Extract the files for WebSphere Business Monitor into the Monitor_installation_image directory. This creates a WBM directory at the root level of the Monitor_installation_image, for example `C:\WBM_Install_Image\WBM`.

3.3.3 Installing WebSphere Business Monitor on a single server

You can use the installation program to install all components of WebSphere Business Monitor on a single server. This scenario is considered a basic installation and is ideal for development test environments, proof-of-concept installations, and simple installation topologies that do not require high availability or failover support.

Before you complete this task, you must complete the following tasks:

- Determine which installation path you are planning to use.
- Prepare the environment for installation.

Archived

Basic installation option

When you select the basic installation path, the installation program installs all WebSphere Business Monitor components, the appropriate version of the required prerequisite software (WebSphere Application Server), and the optional prerequisite software (WebSphere Portal and DB2 Alphablox). During a basic installation, you can choose to use your existing installed database, such as DB2 or Oracle, or install a Derby Embedded database when you are not using portlet-based dashboards with Alphablox features.

After the prerequisite software is set up on the machines, WebSphere Business Monitor is ready to be installed as explained in the following example:

1. Start **launchpad.exe** from the target directory that you specified in the installation preparation wizard utility.
2. In the Launchpad (Figure 3-20), click **Launch the installation wizard for WebSphere Business Monitor**.

Figure 3-20 WebSphere Business Monitor installation Wizard

3. In the Welcome panel (Figure 3-21), click **Next**.

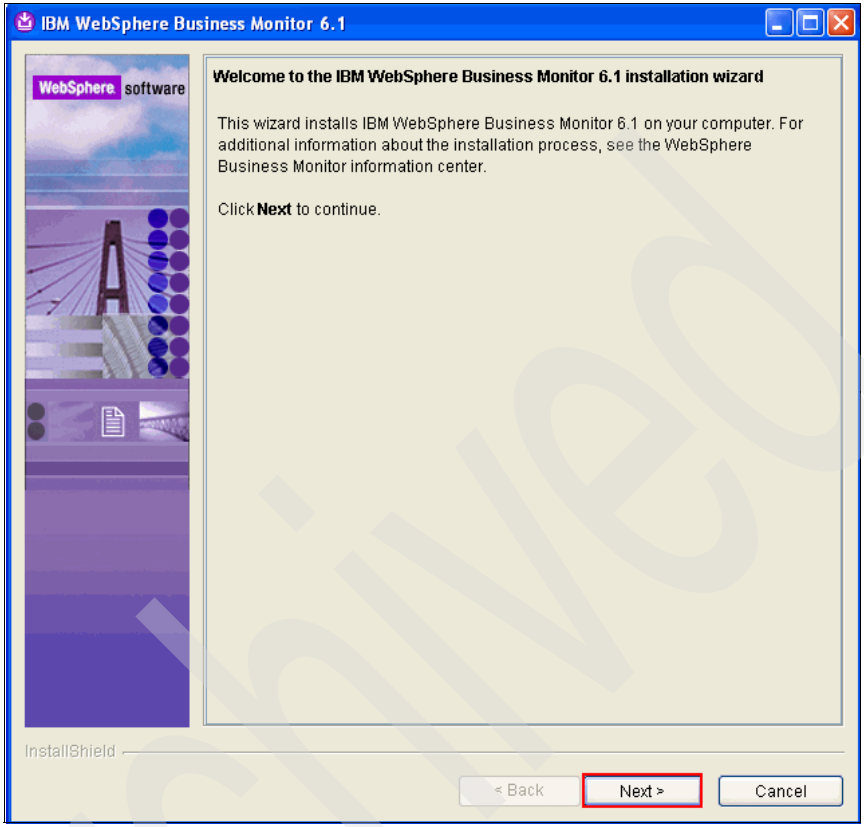


Figure 3-21 Welcome to the IBM WebSphere Business Monitor installation wizard

4. Accept the license agreement and click **Next**.

Archived

5. In the System prerequisites check panel (Figure 3-22), click **Next**.

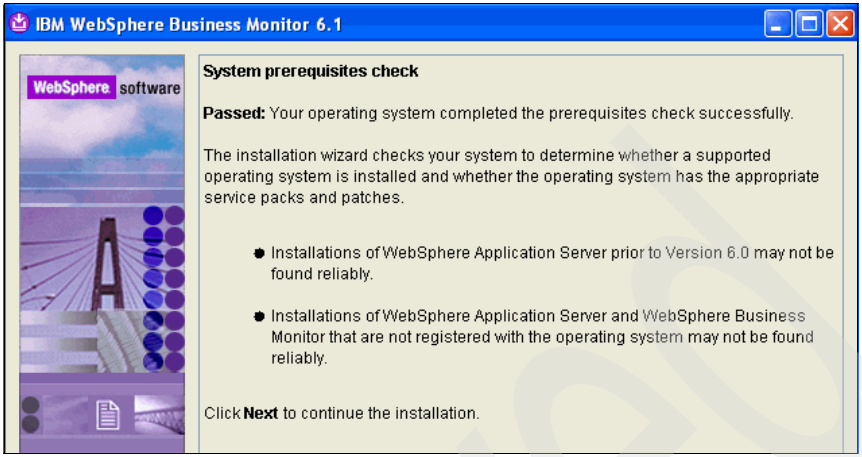


Figure 3-22 WebSphere Business Monitor installation: System prerequisites check

6. In the Installation type selection panel (Figure 3-23), select **Basic Installation** and click **Next**.

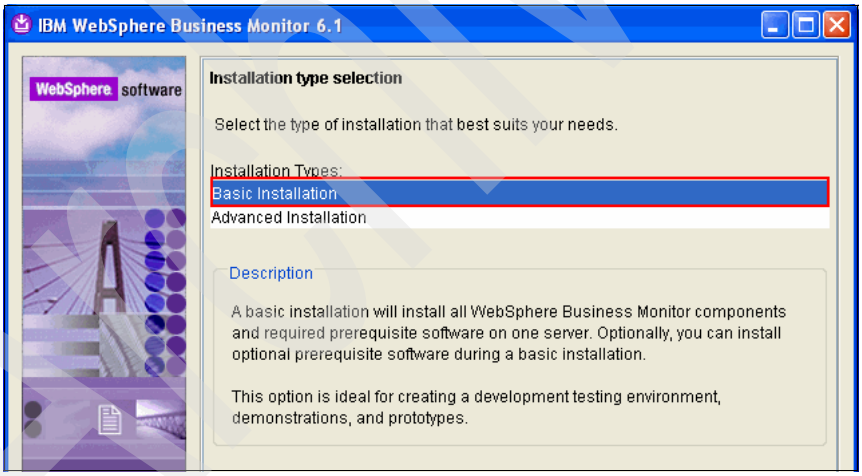


Figure 3-23 WebSphere Business Monitor installation: Installation type selection

7. In the Component selection panel (Figure 3-24), clear the **Portlet-based dashboards with Alphablox features** check box and click **Next**.

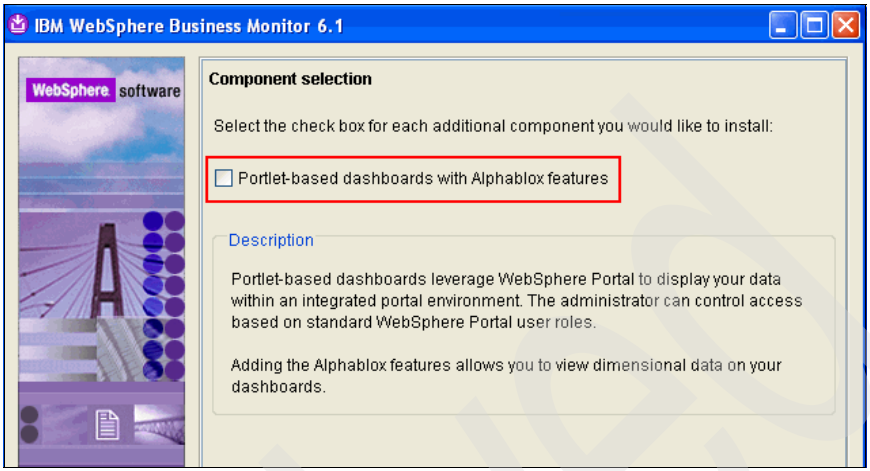


Figure 3-24 WebSphere Business Monitor installation: Component selection

8. In the Installation directory panel (Figure 3-25), type a Business Monitor server installation location. In our example, we enter `C:\IBM\WebSphere\MonServer`. Click **Next**.

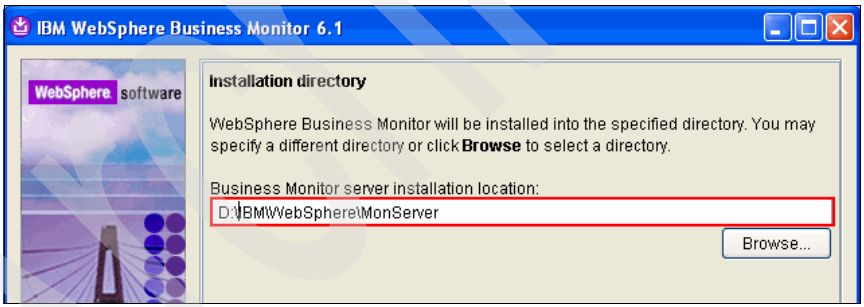


Figure 3-25 WebSphere Business Monitor installation: Installation directory

Archived

9. In the WebSphere Application Server enable administrative security panel (Figure 3-26), clear the **Enable administrative security** check box.

> **Note:** The purpose of this installation is for the test environment and we do not want the security enabled.

Click **Next**.

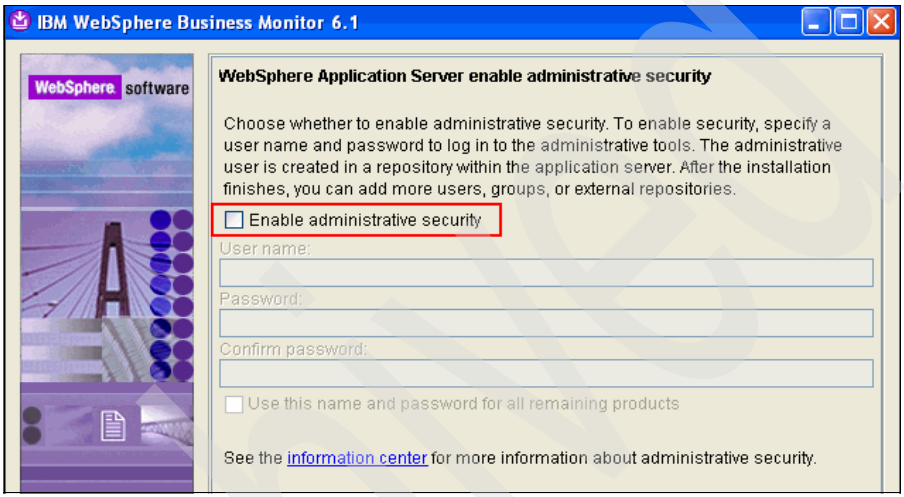


Figure 3-26 WebSphere Business Monitor installation: WebSphere Application Server enable administrative security

Archive

10. In the Human Tasks configuration panel (Figure 3-27), clear the **Configure human task monitor model** check box and click **Next**.

> **Adding Human Tasks to dashboards:** In this installation, we do not plan to use Human Tasks in our dashboards. However, if you want to add Human Tasks to your dashboards:
>
> a. Select the **Configure human task monitor model** check box.
>
> b. In the WebSphere Process Server host name field, enter the fully-qualified host name or the IP address of the server where WebSphere Process Server is installed.
>
> c. In the WebSphere Process Server RMI port number field, enter the port number that WebSphere Process Server uses for remote method invocation (RMI). The default port number is 2809.

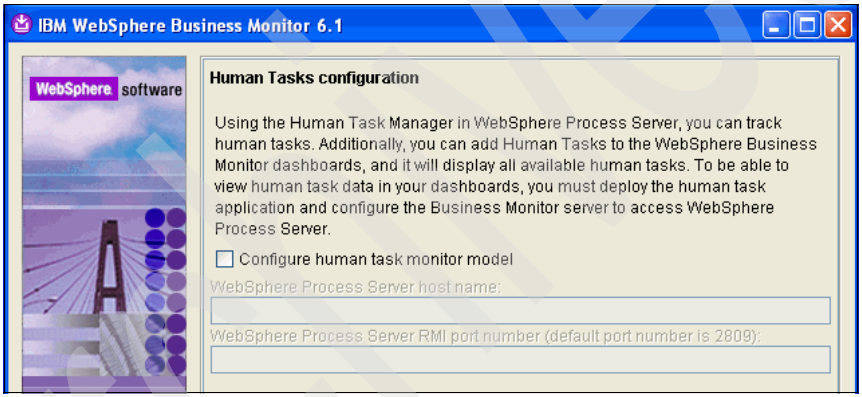


Figure 3-27 WebSphere Business Monitor installation: Human Tasks configuration

11. In the Screen customizations panel (Figure 3-28), click **Next**.

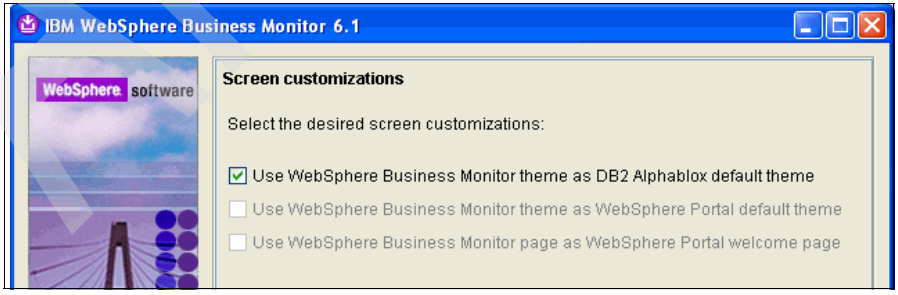


Figure 3-28 WebSphere Business Monitor installation: Screen customizations

12. In the Database configuration panel (Figure 3-29), click **Next**. The new database called MONITOR is created in Derby.

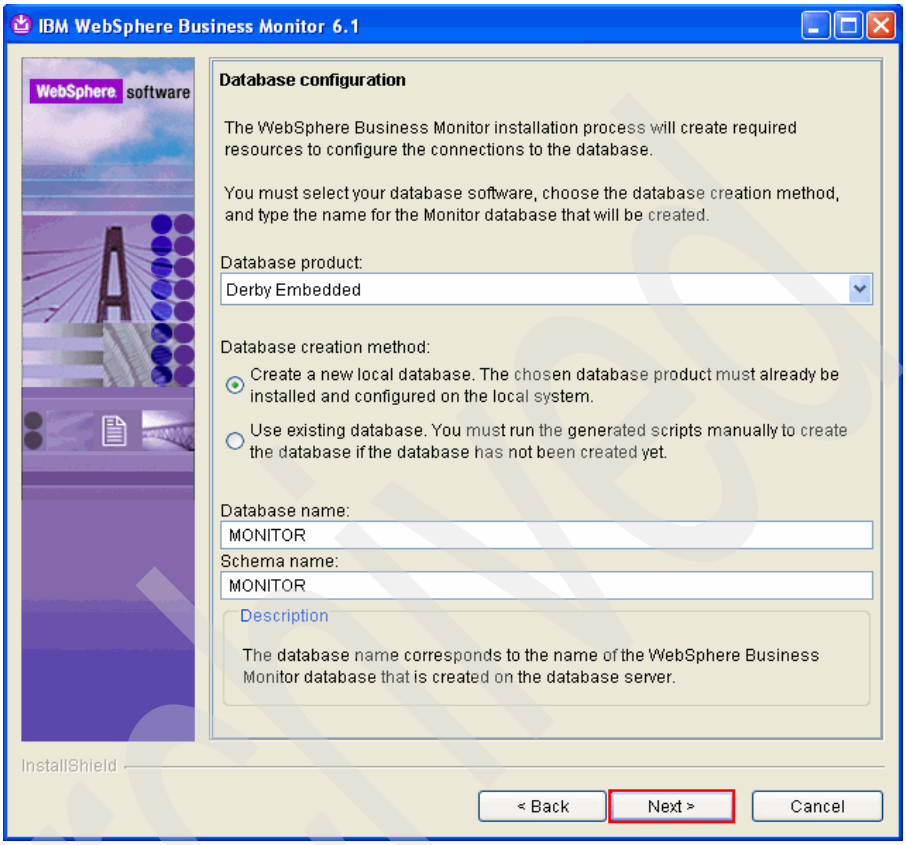


Figure 3-29 WebSphere Business Monitor installation: Database configuration

Archive

13. In the Installation summary panel (Figure 3-30), click **Next**.

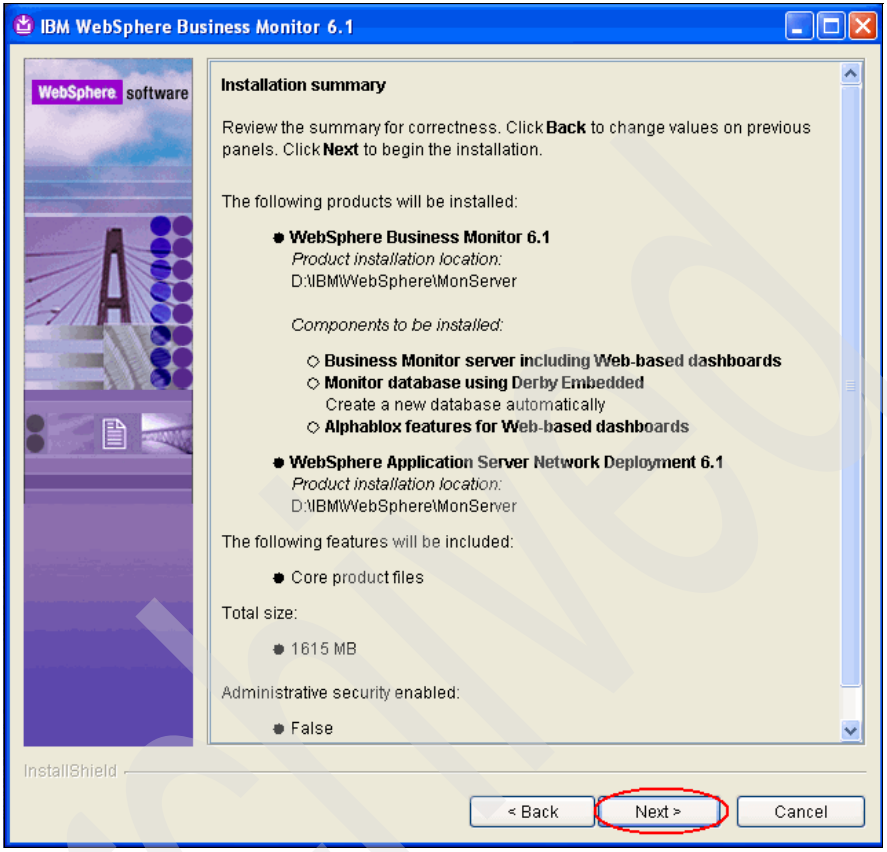


Figure 3-30 WebSphere Business Monitor installation: Installation summary

Archive

14. In the Installation results panel (Figure 3-31), click **Finish**.

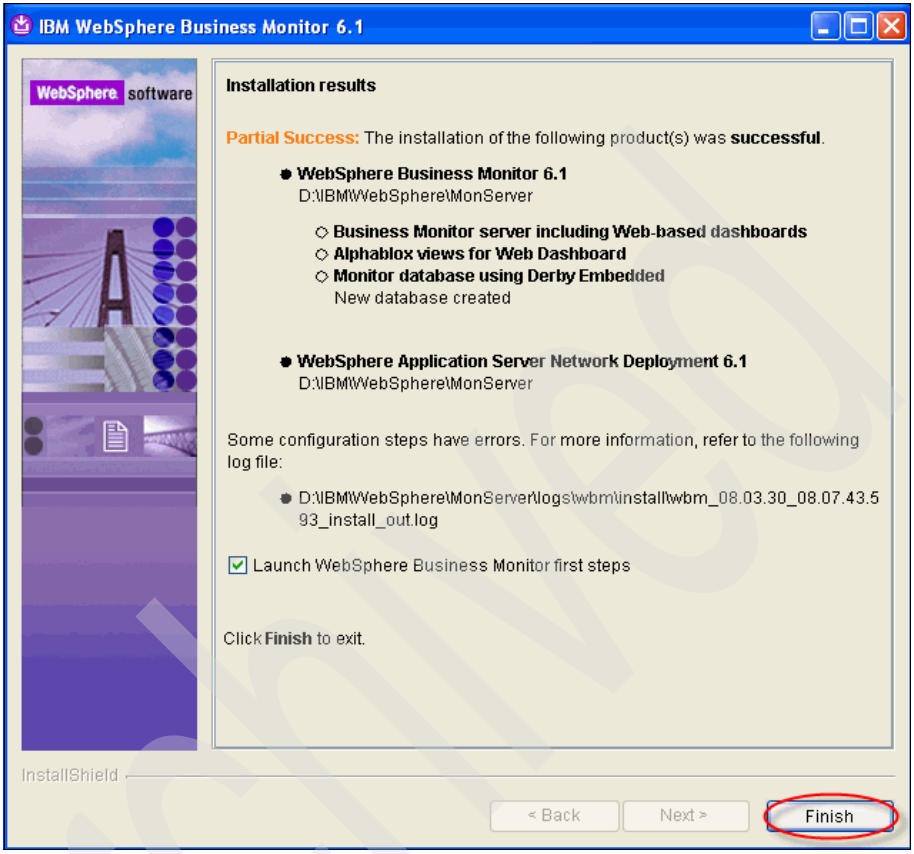


Figure 3-31 WebSphere Business Monitor installation: Installation results

Underscore in host name: If the host name of WebSphere Business Monitor machine has an underscore, then the installation does not create a default profile.

Archived

3.3.4 Verifying the installation

In this section, we explain how to verify the WebSphere Business Monitor installation and how to start and stop the server.

Verifying WebSphere Business Monitor installation

To verify the WebSphere Business Monitor installation:

1. To start the Installation Verification utility, select **Start** → **All Programs** → **IBM WebSphere** → **Business Monitor 6.1** → **Profiles** → **WBMon01** → **First steps**.
2. In the First steps panel (Figure 3-32), select **Installation Verification**.

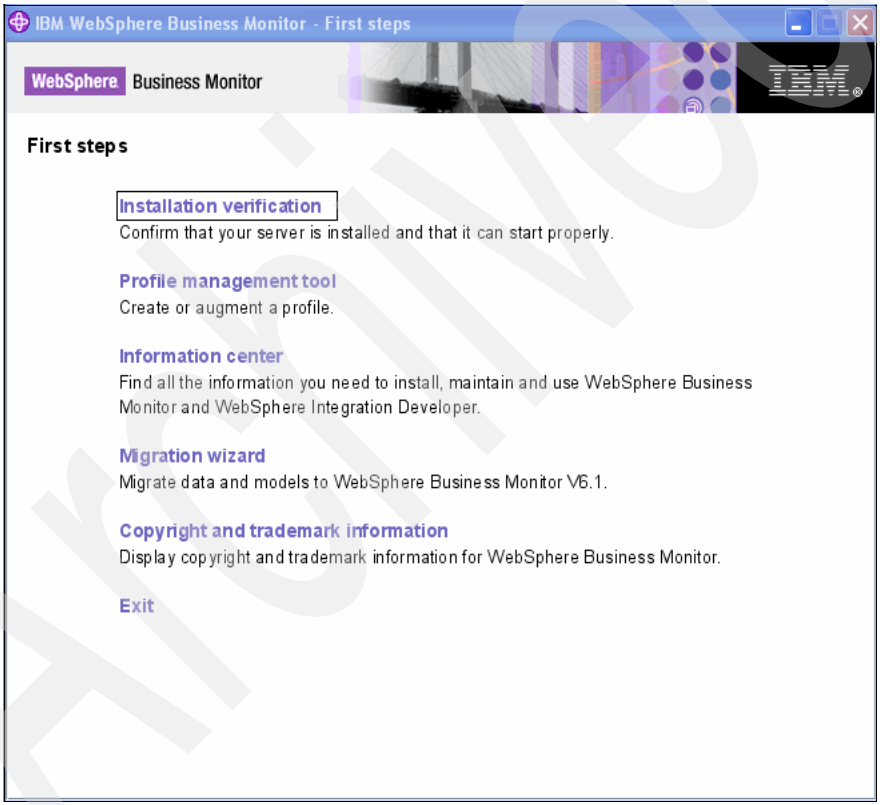


Figure 3-32 Verifying the WebSphere Business Monitor installation

Archive

After the Installation verification completes, the Installation verification panel (Figure 3-33) is displayed.

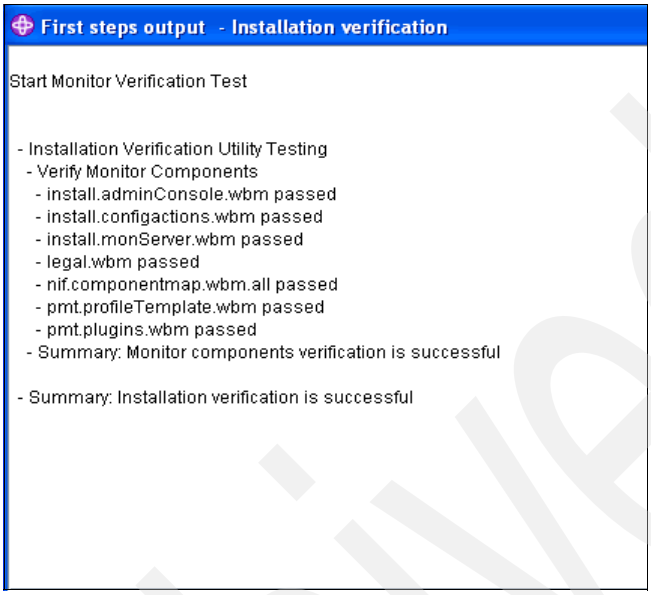


Figure 3-33 Installation verification output

3.3.5 Starting and stopping the WebSphere Business Monitor server

To start the WebSphere Business Monitor server, select **Start** → **Programs** → **IBM WebSphere** → **Business Monitor 6.1** → **Profiles** → **WBMon01** → **Start the server**.

To stop the WebSphere Business Monitor server, select **Start** → **Programs** → **IBM WebSphere** → **Business Monitor 6.1** → **Profiles** → **WBMon01** → **Stop the server**.

Underscore in host name: If the host name of WebSphere Business Monitor machine has an underscore, then the installation does not create a default profile. Therefore, the commands are not available.

Archived

3.4 Deploying and administering monitor models

The WebSphere Business Monitor offers several administrative functions. You can install WebSphere Business Monitor models by using either the WebSphere Administrative Console or scripts.

After you successfully install WebSphere Business Monitor, you administer it through an extension to the WebSphere Administrative Console. In the left pane of the WebSphere Administrative Console under the Applications navigational node, the Monitor Models, Data Movement Service, and Monitor Action Services nodes are displayed (Figure 3-34).

Figure 3-34 WebSphere Administrative Console

By using these navigational nodes, you can access and administer the following different functions by entering your login information (access rights) in the WebSphere Application Server:

- By using the *Monitor Models node*, you can manage your monitor models, including the installation of a specific version of the model on the WebSphere Business Monitor server and the specification of configuration properties.

- By using the *Data Movement Service node*, you can enable a process that periodically copies data from the operational tables to the reporting tables and remove terminated instances from the operational tables.
- By using the *Monitor Action Services node*, you can access the Monitor action services. With the Monitor action services, you can define an action for installing a monitor model.

3.4.1 Installing a monitor model by using the default settings

By using the default settings, you can quickly use the WebSphere Administrative Console to install a new monitor model or to install a new version of an existing monitor model.

Before beginning this task, you must complete the following tasks:

1. Back up the Monitor database.
2. Verify that the name of the monitor model is short.
3. Verify that the path to the WebSphere Business Monitor server is a short path.
4. Log into the administration console.

Windows path-length restrictions: Windows path-length restrictions can be exceeded when long application names are used or when the WebSphere Application Server is installed in a deep directory structure. Keep both the model name and the path as short as possible to avoid possible problems.

If you are installing a new version of an existing monitor model, the version time stamp must be more recent than any previously installed versions of that model. When you install a new version, the previous version is automatically configured to process events that are associated with existing monitoring context instances only (that is Common Event Infrastructure (CEI) distribution mode of active (no new monitoring context instances)). New monitoring-context instances are created only by the new version. Also, events associated with any previous versions that were configured with a CEI distribution mode of inactive (event queue recoverable) are processed by the new version before it processes events on its own event queue.

Archived

To install a new monitor model or a new version of an existing monitor model by using the default settings:

1. From the WebSphere Administrative Console, expand **Applications** and click **Monitor Models**. The table in the right pane lists all currently installed monitor models (Figure 3-35).
2. In the Monitor Models pane, click **Install**.

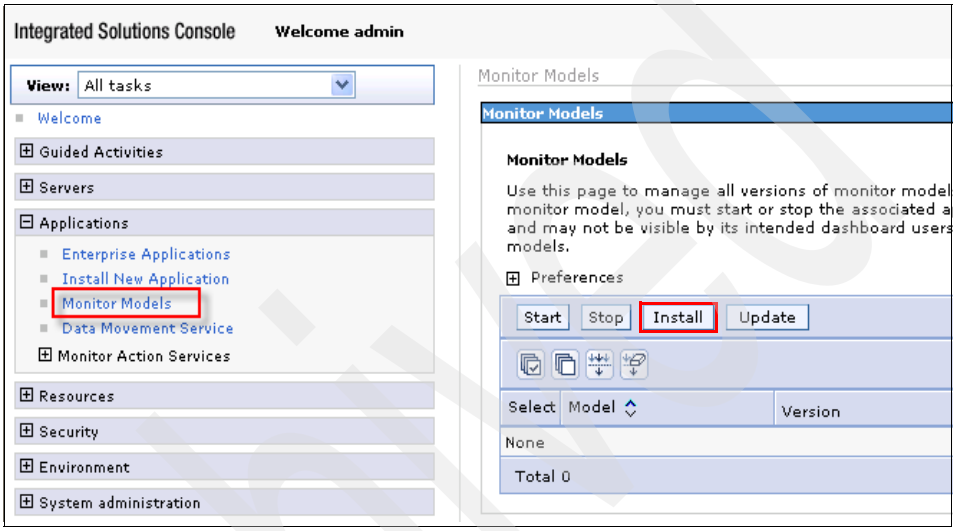


Figure 3-35 WebSphere Administrative Console: Monitor Models

Archived

3. In the next panel (Figure 3-36), for Full path, click **Browse** to specify the location of the monitor model EAR file that you want to install. Under How do you want to install the application?, select **Prompt me only when additional information is required** and then click **Next**.

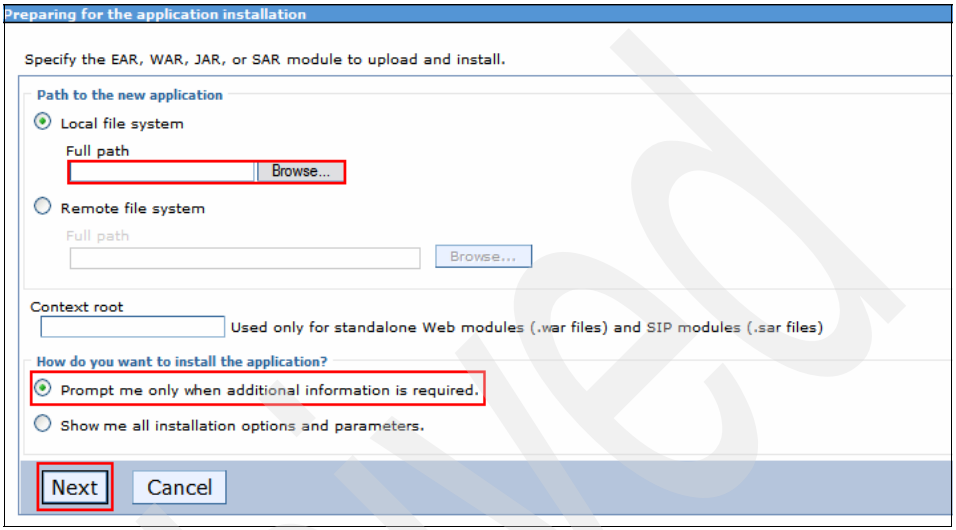


Figure 3-36 Specifying the EAR file to upload and install

Archived

4. In the Step 1: Select Installation options panel (Figure 3-37), click **Next**.

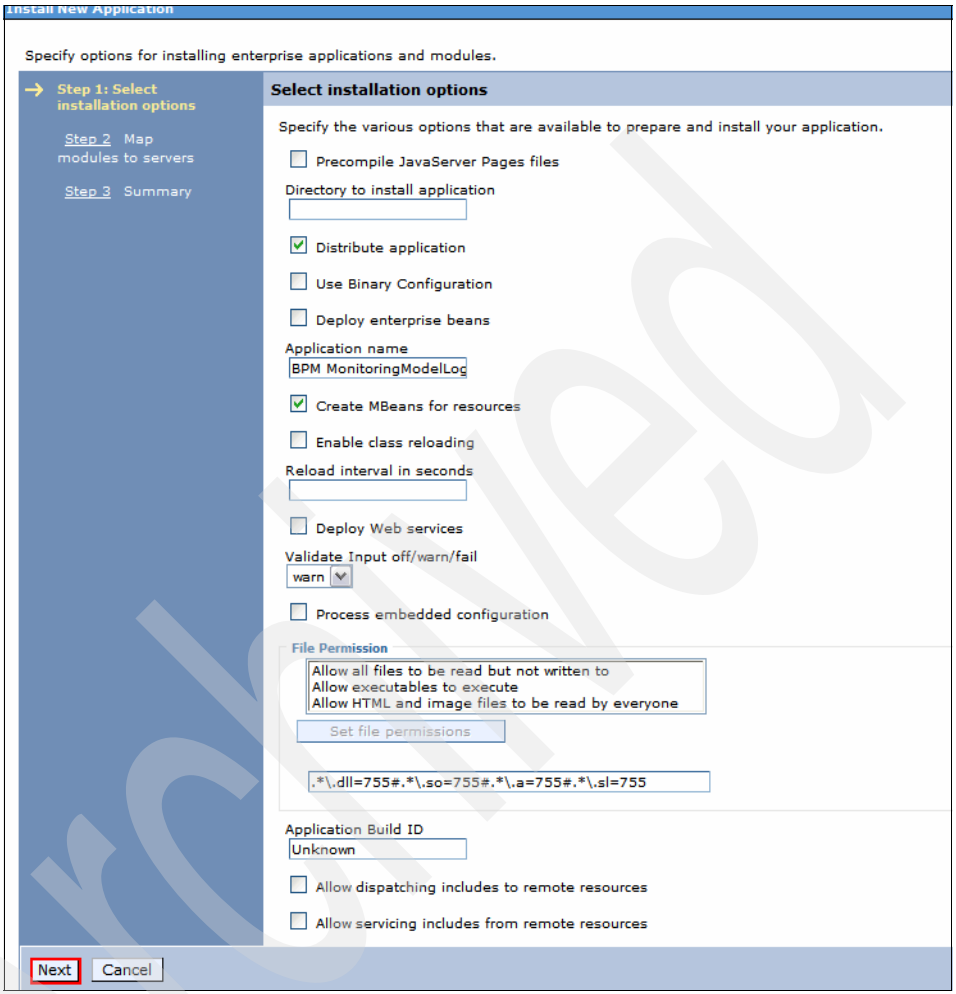


Figure 3-37 Selecting the installation options

Archived

5. In the Step 2: Map modules to servers panel (Figure 3-38), click **Next**.

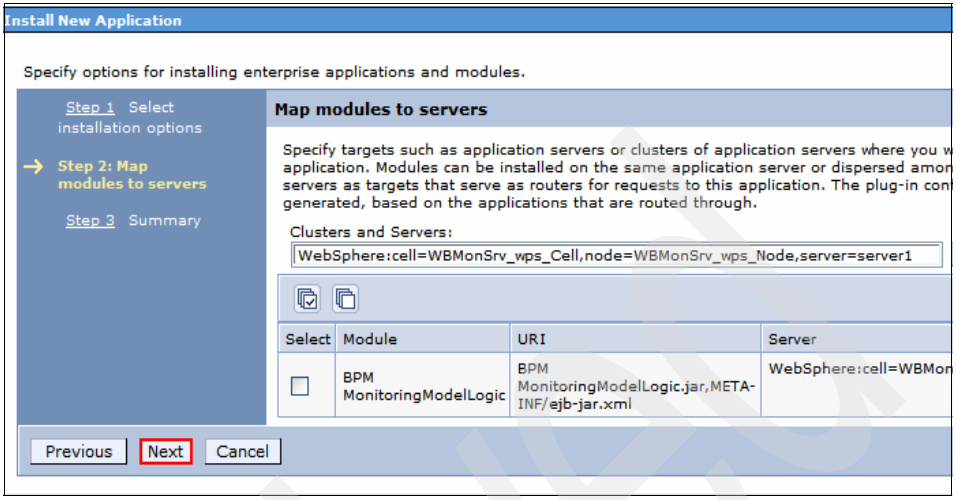


Figure 3-38 Mapping modules to servers

Archived

6. In the Step 3: Summary panel (Figure 3-39), verify that all of the information is correct, and click **Finish**.

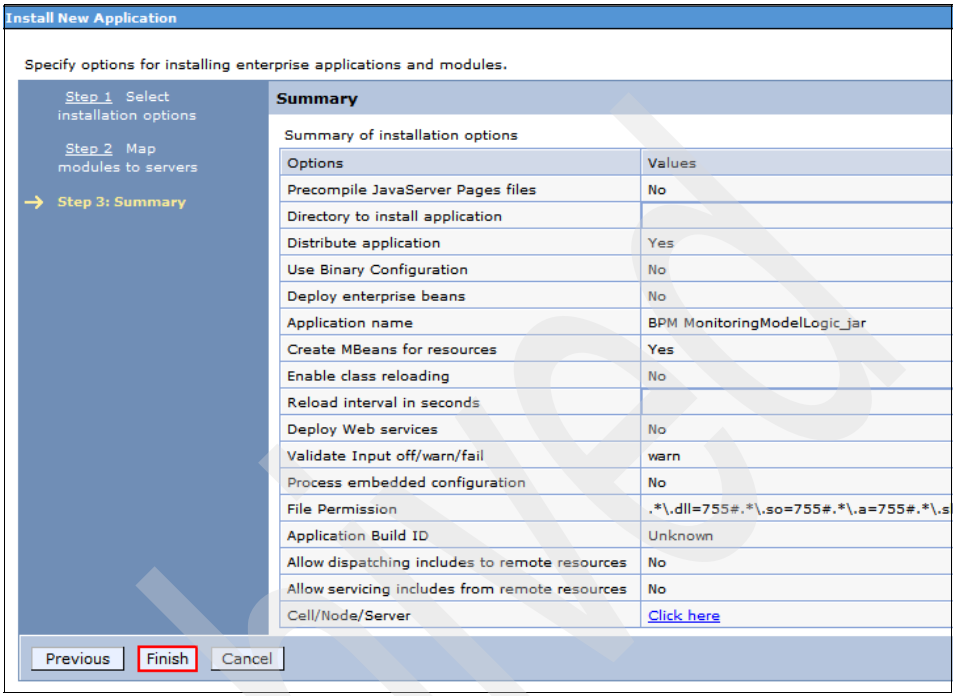


Figure 3-39 Summary of installation options

7. Many informational and warning messages are displayed in the results panel after installing. As long as there are no error messages, click **Review** to review changes before saving or discarding to review, or click the **Save** directly to the master configuration link to save the model (Figure 3-40).

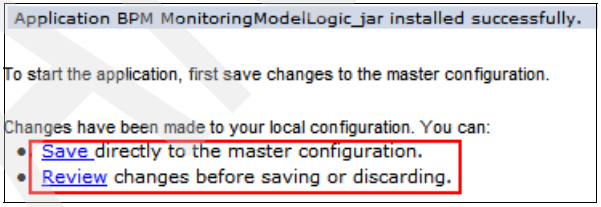


Figure 3-40 Installation result

Archived

3.4.2 Installing a monitor model by using the remote CEI

To install a monitor model with the remote CEI option if the CEI is installed on another machine:

1. From the WebSphere Administrative Console, expand **Applications** and click **Monitor Models**. The table lists all currently installed monitor models in the right pane.
2. In the Monitor Models panel, click **Install**.
3. Specify the location of the monitor model EAR file that you want to install. Click **Show me all installation options and parameters** and then click **Next**.
4. In the Preparing for the application installation panel (Figure 3-41), click **Next**.

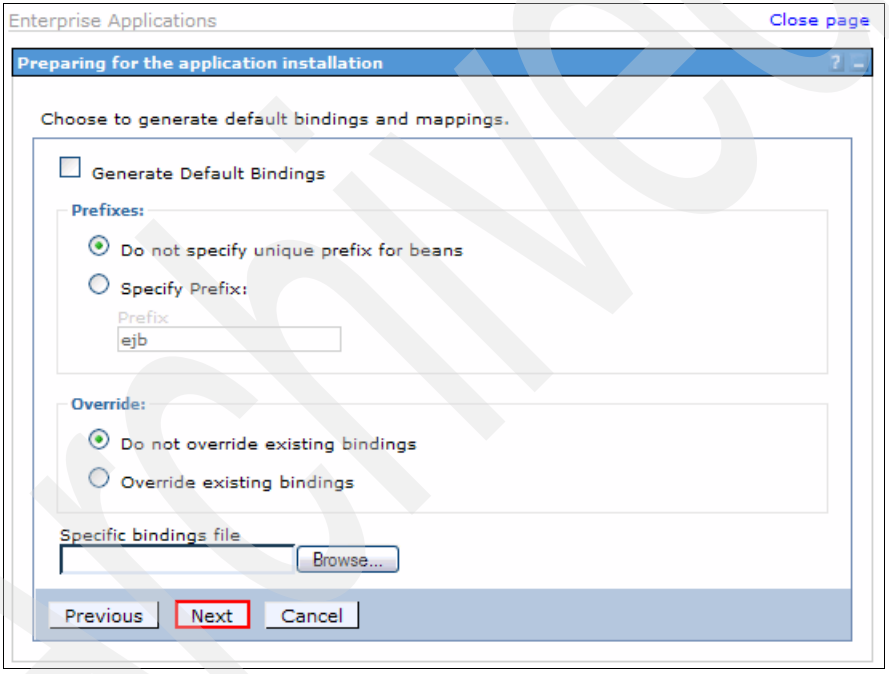


Figure 3-41 Preparing for application installation panel

Archive

5. In the Application Security Warnings panel (Figure 3-42), click **Continue**. This warning message is for informational purposes.

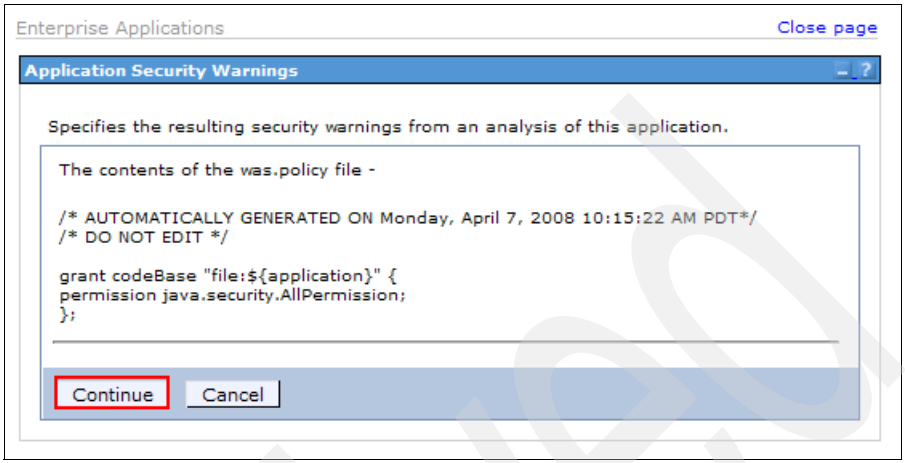


Figure 3-42 Application Security Warnings panel

Archived

6. In the Install New Application panel (Figure 3-43), in the left navigation pane, click **Step 11 Select Monitor model CEI options**.

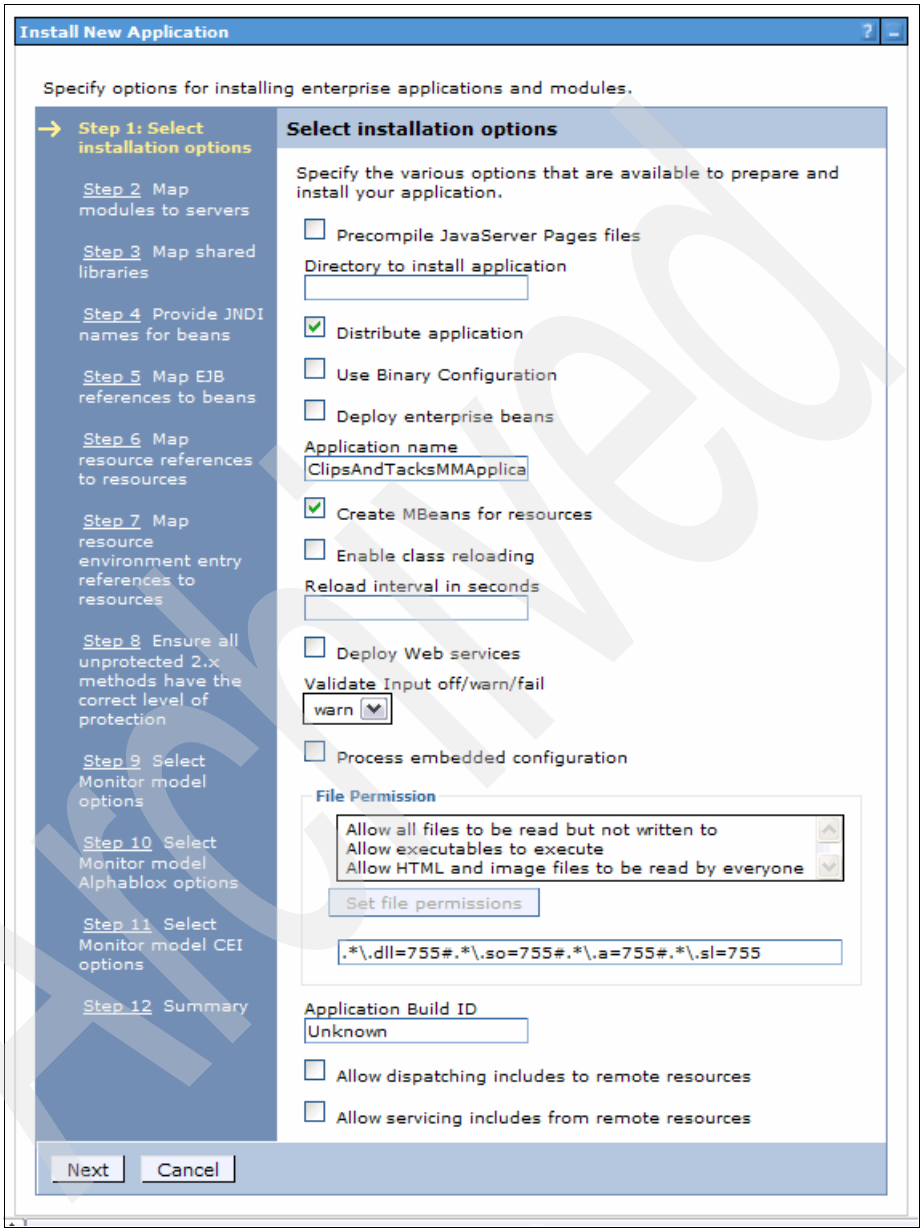


Figure 3-43 Install New Application panel

Archived

7. In the Step 11 Select Monitor model CEI options panel (Figure 3-44 on page 101):
   a. In the Host name field, enter the host name of the server where the CEI is installed. The host name may be a local host if the CEI server is installed on the same physical machine as the WebSphere Business Monitor server.
   b. In the RMI port field, enter the RMI port number of the server where CEI is installed. RMI port is the BOOTSTRAP ADDRESS port of the Application Server, and the default is 2809.
   c. In the Security field, select either **Disabled** or **Enabled**. If you select Enabled, enter the user ID and password for this server.

      Enabled security: If you enable security, Administrative security must also be enabled on the WebSphere Business Monitor server. Otherwise, the CEI configuration fails.

   d. Select the event-group profile list name from the table. If this location is remote, ensure that the host name, the RMI port, and security, if required, are entered, and click **Refresh List** to see the event-group list names for that server.
   e. For Distribution Mode, select Active or Inactive depending on whether you want to configure CEI during installation to distribute events. CEI can be configured later.
   f. Click **Next**.

Archived

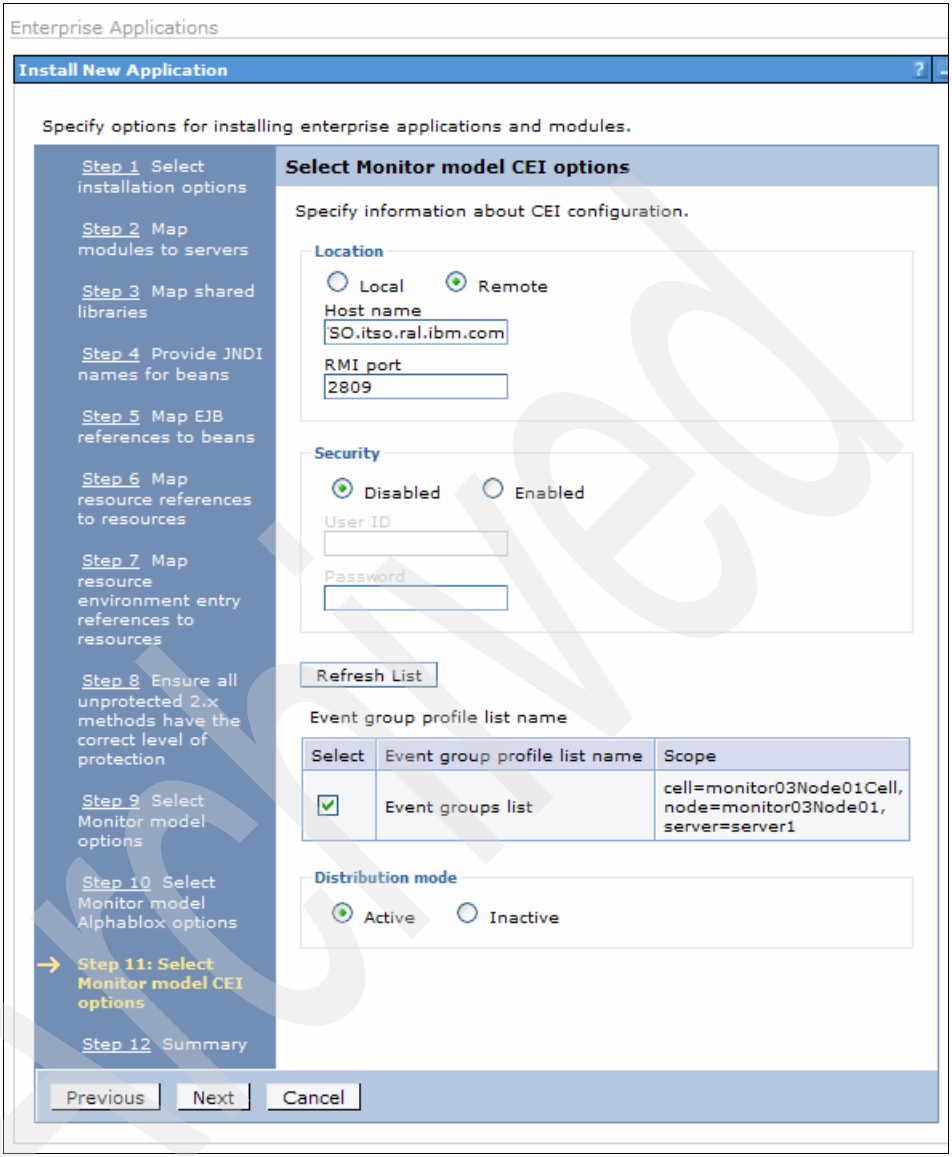


Figure 3-44 Select Monitor model CEI options panel

Archived

8. In the Summary panel (Figure 3-45), verify that all of the information is correct, and click **Finish**.

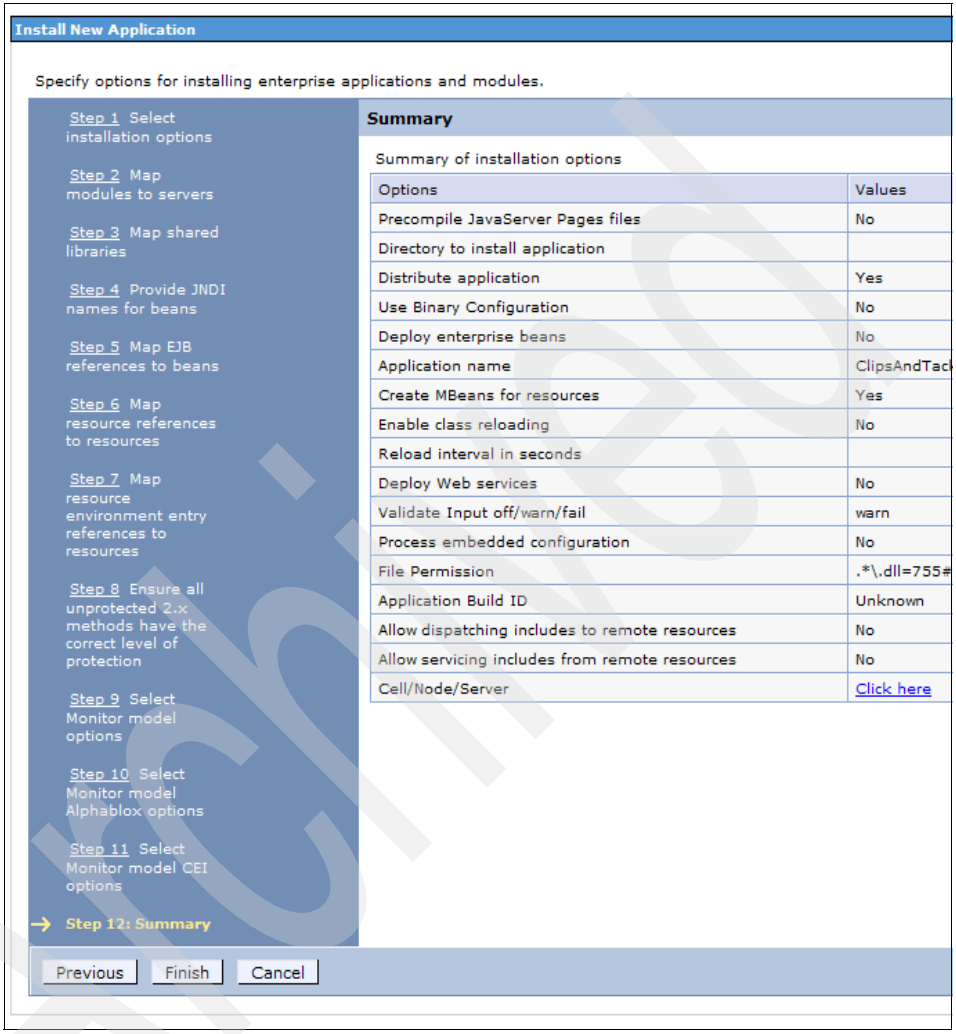


Figure 3-45 Summary

9. Many informational and warning messages are displayed in the results panel after the installation. As long as there are no error messages, click **Review** to review changes before saving or discarding or click **Save** to save the model directly to the master configuration.

Archived

3.4.3 Starting and stopping a monitor model

You can use the WebSphere Administrative Console to start or stop a version of an installed monitor model.

Login: Before you begin this task, you must log in to the WebSphere Administrative Console.

To start or stop an installed monitor model:

1. In the WebSphere Administrative Console (Figure 3-46), expand **Applications** and click **Monitor Models**.

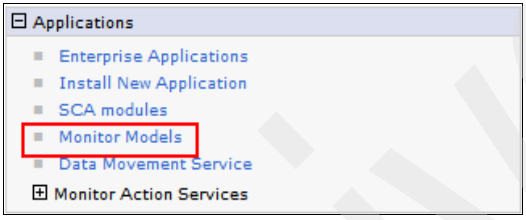


Figure 3-46 WebSphere Administrative Console

2. In the Monitor Models panel (Figure 3-47), in the Select column, select the model and then click **Start** or **Stop** as appropriate. The icon in the Status column is updated to reflect this action.

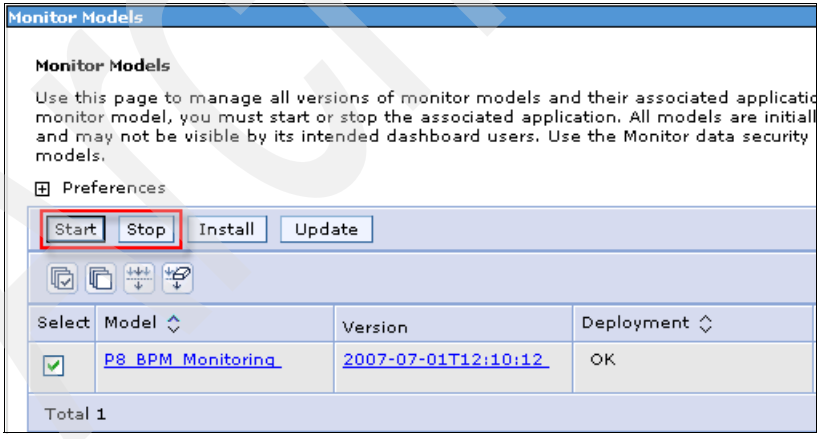


Figure 3-47 Selecting the monitor model

Archived

3.4.4 Versioning a monitor model

Monitor models can be versioned to fix problems or add enhancements. *Versioning* refers to the situation where a user modifies aspects of a monitor model to fix problems or make enhancements (adds a metric, monitoring context, and so on). The user wants to deploy this new version of the model for future event processing and dashboard reporting, while preserving the data collected from previous model versions.

You create the monitor model versions by using the Monitor Model editor. To version a model, update the model version time stamp, but leave the model ID as it is. If you update the model ID, WebSphere Business Monitor considers it a new model (Figure 3-48). The Monitor Model editor produces an enterprise archive (EAR) file for each model version that is created. The project names used during EAR generation must be unique per model version. You can then install the model versions by using the WebSphere Administrative Console, in the same fashion that a new model is deployed.

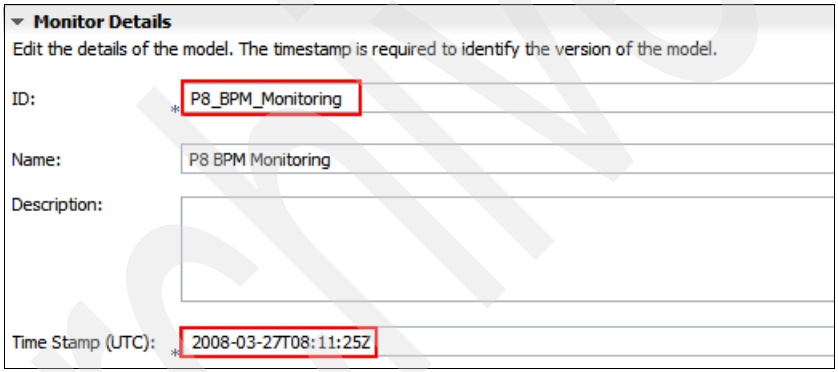


Figure 3-48 Monitor Details

A monitor model can have multiple versions installed but only the most recent version can have a CEI distribution mode of *Active*. Any number of previous versions, however, can have a CEI distribution mode of *Inactive* (no new monitoring context instances). This means that events related to new monitor context instances go to the new version, while events relating to existing monitoring context instances go to the old version.

When the user deploys a new version of the monitor model, a set of tables and views is created in the Monitor database to support that version. Additionally, a set of cross-version views is created to support dashboard queries that require data across all the current and previous model versions. When reporting on data that did not exist in previous model versions, null values are returned. Because the database views union data together across model versions, it is not possible

to change the data type of an existing metric. All other monitor model changes are supported.

To deploy a new version of a model as an entirely independent model instead, you must first change the ID of the model. In addition, you must also change the names of all the J2EE projects generated during EAR generation to be unique, so that they do not collide with the names that are used by the deployed model. If you deploy a model as an independent model rather than as a new version, no tables or data is shared between the models, which is the case for any other two independent models deployed on the WebSphere Business Monitor server.

Because the data movement service always works on a version-specific basis, no special user intervention is needed for versioning when the data movement service is enabled.

Installing a new version of an existing model

You can install a new version of an existing model when one or more of the previous versions have a CEI distribution mode of *Inactive* (event queue recoverable). When a new version of an existing monitor model is installed on a production mode server, all active monitoring context instances from the previous version must be moved to the new version. Then, the CEI distribution mode of the new version can be set to *Active* if any previous versions have a CEI distribution mode set to *Inactive* (event queue recoverable).

Checking the number of active monitoring context instances

Check the number of active monitoring context (MC) instances for the previous version on the Version Details page for the previous version in the WebSphere Administrative Console:

1. From the WebSphere Administrative Console, click **Application** → **Monitor Models**. The table lists all currently installed monitor models.
2. In the Monitor Models panel (Figure 3-49 on page 106), click the previous version.

Archive

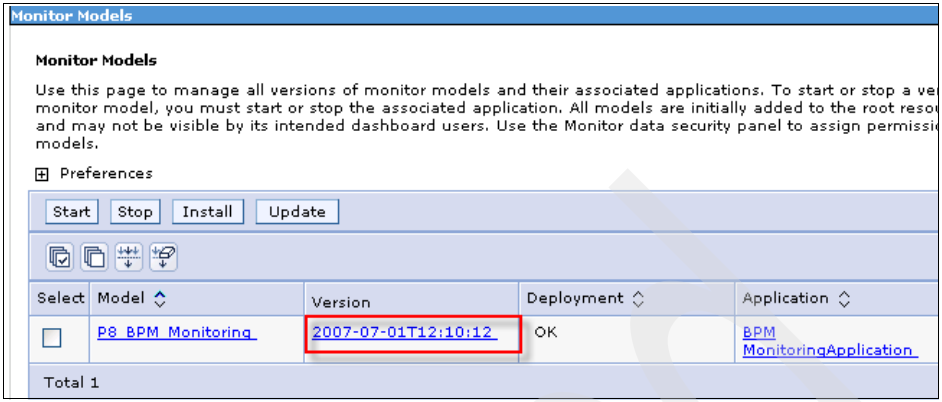


Figure 3-49 Monitor Models

The Version Details panel (Figure 3-50) shows the Active MC instances.

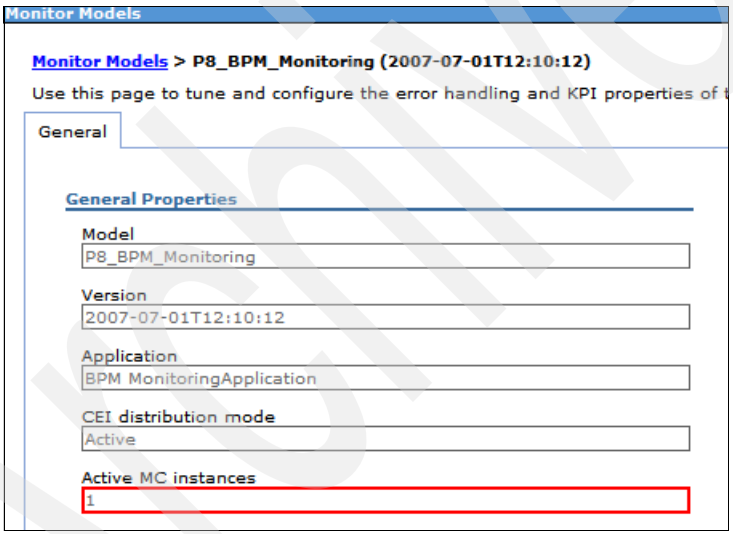


Figure 3-50 Active MC instances

Running the LifecycleServices MBean method

If the number of Active MC Instances is greater than 0, then invoke the following LifecycleServices MBean method before setting the CEI distribution mode of the new version to **Active**:

```
LifecycleResultsBean moveMCInstances(String modelID, long versionDate,
long toVersionDate, boolean activeInstancesOnly)
```

Archive

You can invoke this method by using the **wsadmin** utility:

```
set ls [$AdminControl completeObjectName type=LifecycleServices,*]
$AdminControl invoke $ls moveMCInstances {"modelID", versionDate,
toVersionDate, true}
```

In this command, *model_ID* is ID of the model. *versionDate* is the previous version with a CEI distribution mode set to Inactive (event queue recoverable). *toVersionDate* is the new version.

If the **moveMCInstances** command fails and you can manually move the active monitoring context instances via other means or you ignore the active monitoring context instances, you can invoke the following LifecycleServices MBean method:

```
LifecycleResultsBean confirmMoveMCInstances(String modelID, long
versionDate, long toVersionDate)
```

Changing the CEI distribution mode to Active

After the command successfully completes for each previous version with active monitoring context instances, you can change the CEI distribution mode of the new version to Active:

1. From the WebSphere Administrative Console, click **Application** → **Monitor Models**. The table lists a currently installed monitor models.
2. Click the new version.
3. In the Monitor Models panel (Figure 3-51), under Version Properties, click **Change CEI distribution mode**.

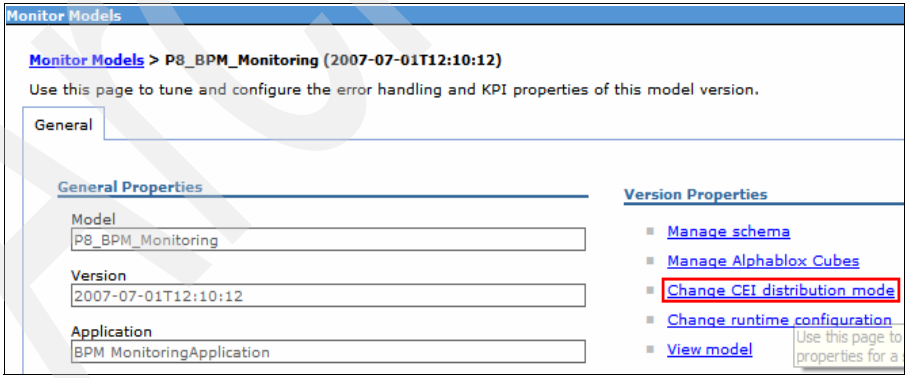


Figure 3-51 Version details

Archived

4. On the General page (Figure 3-52), for Target, select **Active** and click **OK**.

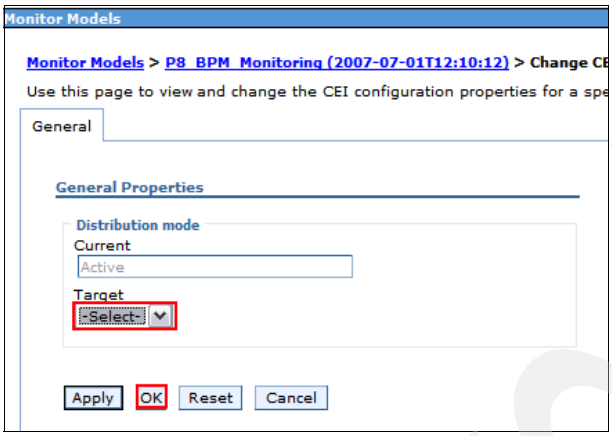


Figure 3-52 Changing the CEI distribution mode

3.4.5 Updating a monitor model

You can use the WebSphere Administrative Console to update a version of a monitor model to replace the application code associated with the model. You can update a specific version of a monitor model only to replace the application code associated with the model, for example, when the application file is regenerated after applying an interim fix to correct a code-generation error. If you modified the monitor model (for example, by adding or removing a metric, you must install a new version of the monitor model.

Login: Before you begin this task, you must log in to the WebSphere Administrative Console.

Archived

To update an installed monitor model:

1. In the WebSphere Administrative Console, expand **Applications** and click **Monitor Models**.
2. In the Monitor Models panel (Figure 3-53), in the Select column, select the model that you want to update. Click **Update**.

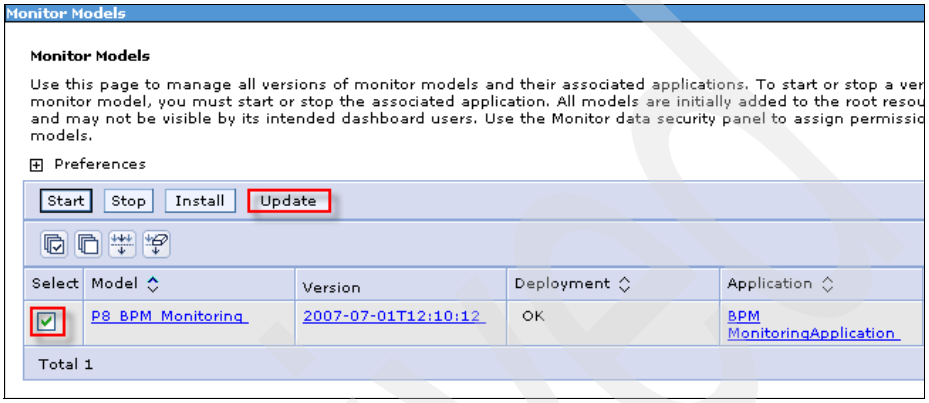


Figure 3-53 Monitor Mode panel

3. In the Preparing for the application installation panel, specify the location of the monitor-model EAR file that you want to update, and click **Next**.
4. Complete the steps in the WebSphere Application Server installation wizard and save the changes to the master configuration. You might have to start this version of the model again.

3.4.6 Removing a monitor model and data

You can remove a monitor model in several ways. In this section, we explain how to uninstall a single version, uninstall all versions, and purge the model after all versions are uninstalled. You can also remove the data that is associated with a model in various ways. For example, you can remove the data when the model is uninstalled. Another way is to separately remove the data either automatically or by using a script, which we explain in this section.

Uninstalling a single version of a monitor model

Uninstalling a single version of a monitor model removes only the specified version of the model from the database. To completely remove a model, all versions of a model must be uninstalled.

To uninstall a single version of a monitor model:

1. In the WebSphere Administrative Console, expand **Applications** and click **Monitor Models**.
2. In the Monitor Models panel (Figure 3-54), in the Select column, select the check box next to the applicable model. Then click **Stop**.

   In the Version column, click the appropriate version.

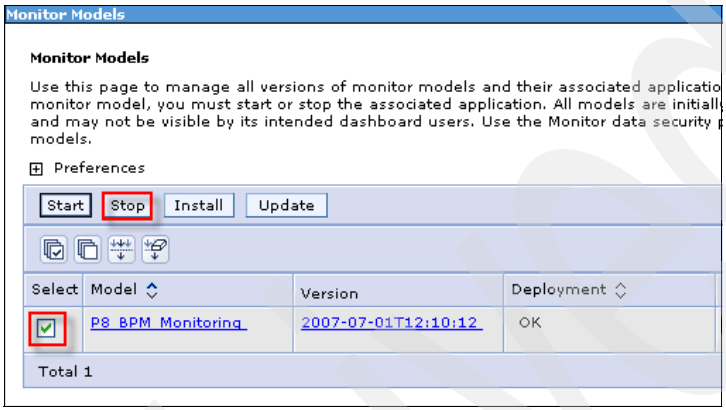


Figure 3-54 Monitor Models

Archive

3. (Optional) If you want to delete the schema (meaning, the data for dashboard reporting) during uninstallation, click **Manage Schema**. In the Manage schema panel (Figure 3-55), select **Run script to delete the schema during uninstallation** and then click **OK**.

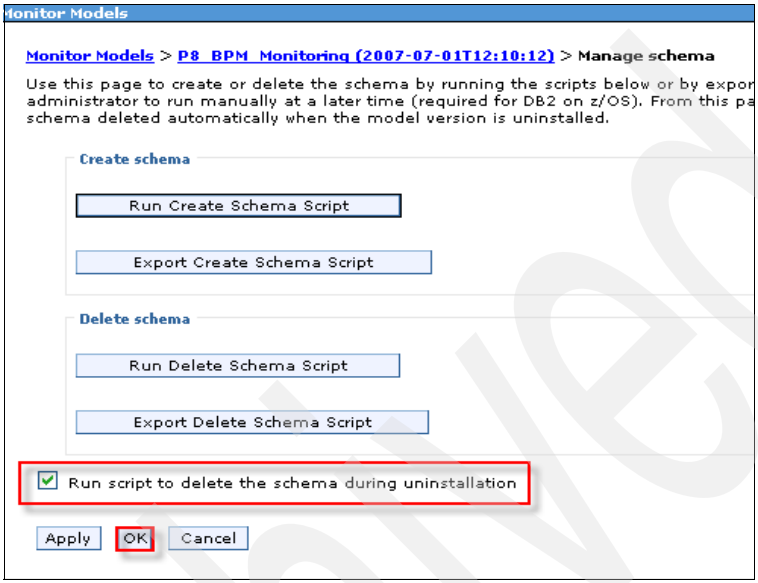


Figure 3-55 Monitor Models

4. Click **Applications** → **Enterprise Applications**. All of the installed applications are displayed.

Archived

5. In the Enterprise Applications panel (Figure 3-56), in the Select column, select the version of the monitor model you want to uninstall. Click **Uninstall**. To confirm the uninstallation, click **OK**.

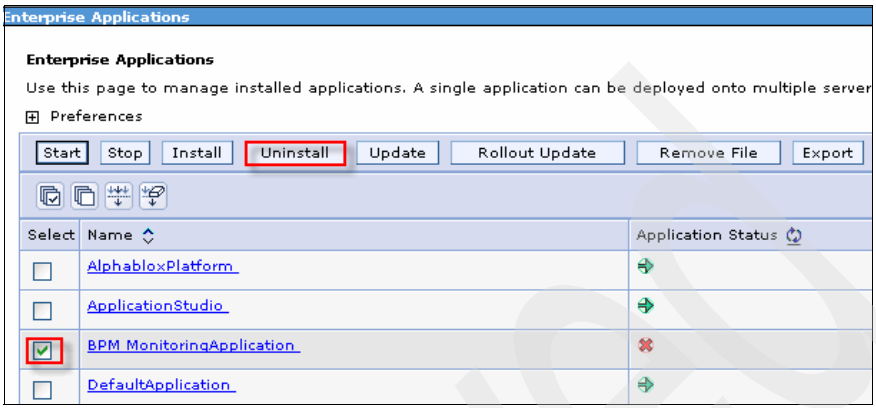


Figure 3-56 Enterprise Applications

6. Click **Save**.

Uninstalling all versions of a monitor model

To completely uninstall a model from the database, you must uninstall all versions of a model as described in the previous section.

Purging a monitor model

Purging a monitor model removes all versions of a model from the repository database. It also removes the monitor model definition from the Monitor database. Purging is only necessary after you follow all of the instructions for uninstalling all versions of a monitor model. The model is still displayed on the Monitor Models page in the administrative console.

Before you purge a monitor model, all versions of a model must be uninstalled.

Attention: Purging a monitor model removes the database schema, which means all historical data for that model is deleted.

To purge a monitor model that was installed in WebSphere Business Monitor:

1. In the WebSphere Administrative Console, expand **Applications** and click **Monitor Models**.
2. In the Monitor Models panel, in the Model column, click the name of the monitor model you want to purge.

3. On the General page (Figure 3-57), under Model Properties, click **Purge**.

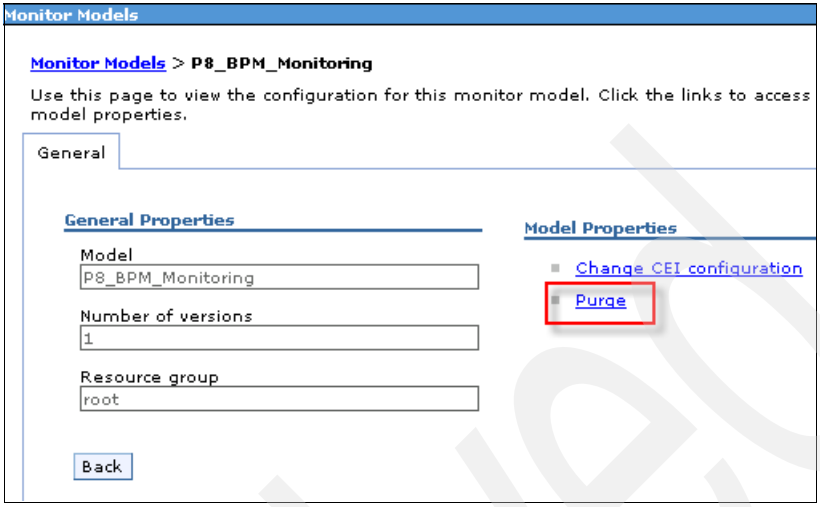


Figure 3-57 Monitor Models

4. To confirm the deletion, click **Purge** (Figure 3-58). The purge action removes all versions of the selected monitor model.

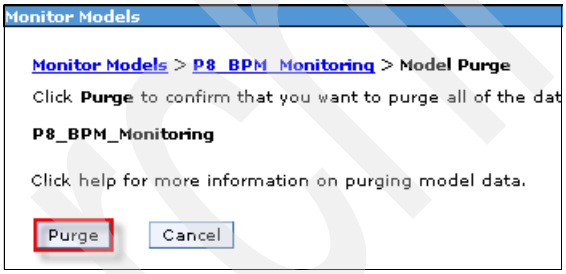


Figure 3-58 Model Purge

Running the delete schema scripts automatically

You can use the WebSphere Administrative Console to run the delete schema scripts automatically.

Attention: Deleting schema scripts erases all the data associated with a model.

Archived

To run the delete schema scripts automatically:

1. In the WebSphere Administrative Console, expand **Applications** and click **Monitor Models**.
2. Click the version of the model that you want to manage. Click **Manage Schema**.
3. In the Manage schema panel (Figure 3-59), click **Run Delete Schema Script**.

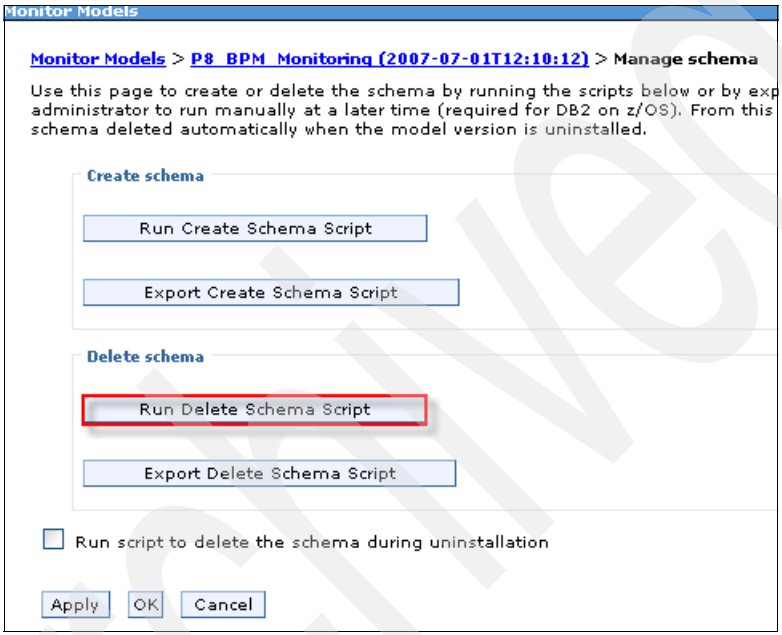


Figure 3-59 Run Delete Schema Script

Exporting delete schema scripts

To export delete schema scripts to a DDL file, which can be run later or remotely:

1. In the WebSphere Administrative Console, expand **Applications** and click **Monitor Models**.
2. Click the version of the model you want, and then click **Manage Schema**.
3. Click **Export Delete Schema Script**.
4. Save the file in an appropriate location.

This procedure produces a script that you can run manually later by using a command prompt or from a remote location.

Archived

3.5 Monitoring with Web-based dashboards

WebSphere Business Monitor 6.1 offers two dashboard deployment choices:

- A portal-based dashboard
- A Web-based dashboard (without requiring portal)

Both dashboards provide identical monitoring capabilities. Web-based dashboards, which we discuss in this book, are a component of WebSphere Business Monitor version 6.1 running on WebSphere Application Server version 6.1.

Dashboards are implemented as Web pages. A dashboard that displays specific business performance consists of a variety of ways to view your data.

After you create your dashboard, use the utilities to create key performance indicators (KPIs), export values, and subscribe to alerts. Additionally, you can add one of the following items to your dashboard:

- *Alerts* to display alerts that notify users of defined situations occurring at run time
- *Diagrams* to display diagrams that are associated with a particular monitoring context or KPI context
- *Human Tasks* to display all available human tasks and to perform actions on them

  This item is supported for WebSphere Process Server V6.1 only.
- *Instances* to display the available monitoring contexts in either individual instances or user-defined groups of context instances
- *KPIs* to display details of KPIs, such as a KPI value relative to the defined ranges and the target, if applicable, and the status

Optionally, you can install DB2 Alphablox, version 9.5 so that users can add the following items on their dashboards:

- Dimensions

  Use Dimensions to display a multidimensional view of business performance data. You can generate multidimensional reports that analyze different aspects of data retrieved from a multidimensional data mart.
- Reports

  Use Reports to display performance reports relative to a time axis. This item helps you to analyze historical values of numeric business measures that belong to a cube.

Archive

The three following WebSphere Business Monitor utilities are supported:

- Alerts Subscription

  Use this utility to subscribe and unsubscribe to different predefined alerts. Along with subscribing to an alert, the user can choose the type of notification channel associated with each alert.

- Export Values

  Use this utility to export data resulting from a specific monitor model and duration to an XML file that can be imported by WebSphere Business Modeler.

- KPI Manager

  Use this utility to define, copy, and update KPIs directly from the dashboard interface.

3.5.1 Modes in your dashboard

Items in your dashboard have the following modes, which you can use to prepare, change, and use your dashboards to best suit your needs:

- View mode

  By using View mode, you can view the data content and perform actions to manipulate the data. These actions depend on the nature of the data and the way that it is represented (that is, in tables or charts). For example, you can filter data or sort data, select date ranges for the report you are generating, select the frequency of the report generation, drill down certain data, and so forth.

  If security is enabled, the data viewed depends on the role and access privileges of the user viewing them. Only administrators can view all the data in the system using dashboards unless users have specifically been given access rights.

- Personalize mode

  The Personalize mode consists of a group of the settings that is normally limited to customization of visual properties. However, you can also use Personalize mode to configure the items on your dashboard. All WebSphere Business Monitor dashboard items and the Alerts Subscription utility support a Personalize mode. However, the Export Values and KPI Manager utilities do not.

Archived

3.5.2 Creating a dashboard

A *dashboard* is a Web page that can contain one or more items that graphically represent business data. An administrator or a user with the required access permission can create a dashboard and add data to it.

To create a dashboard:

1. To launch the WebSphere Business Monitor dashboard, open a browser and type the following URL:

   `http://[host]:9080/BusinessDashboard`

2. Log in to the WebSphere Business Monitor Dashboard (Figure 3-60).

Figure 3-60 WebSphere Business Monitor Dashboard

The Getting Started main page is displayed. In WebSphere Business Monitor V6.1, the end-user experience has been revamped with an improved look and feel for higher responsiveness and ease of personalization. The Getting Started Guide provides a collection of videos and instructional guidance, which is a convenient way to learn what you can accomplish by using dashboards.

Archived

Figure 3-61 shows the Getting Started main page with the content areas to explore.

Getting Started guide: The Getting Started guide includes learning modules for eight of the most common tasks for end users with line of business responsibilities. It also demonstrates how power users can customize the business activity monitoring (BAM) experience and create new KPIs. Users can navigate Getting Started through the tabs at the top of the page or by clicking the icons in the content areas.

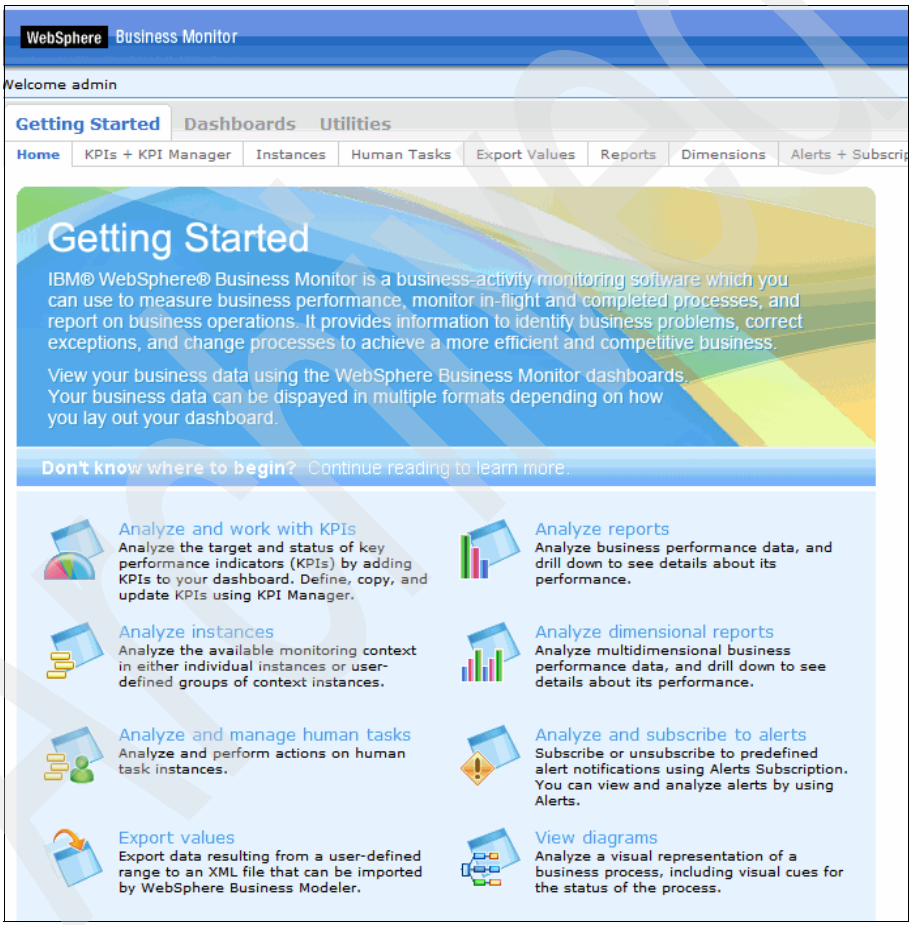


Figure 3-61 Getting Started

3. Click the **Dashboards** tab (Figure 3-62).

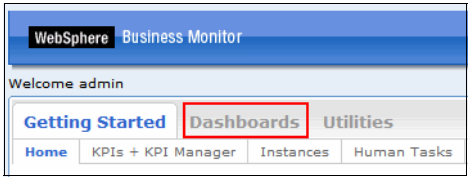


Figure 3-62 Select Dashboards tab

4. On the Manage page (Figure 3-63), click the **New** button to create a dashboard.

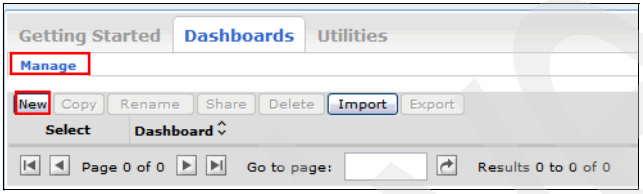


Figure 3-63 Manage Dashboard

5. In the New Dashboard panel (Figure 3-64):
   a. In the Name field, type the name of the new dashboard. For example, you might type the name `Exclusive`.
   b. From the Layout list, select a layout.
   c. Click **OK** to save and open the new dashboard.

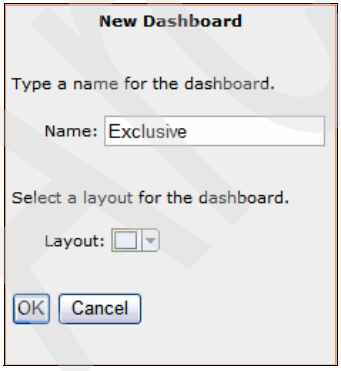


Figure 3-64 New Dashboard window

Archived

Adding to your dashboard

You can add one or more items to your dashboard to monitor aspects of business performance. To add data, choose either of the following options:

- From a dashboard using the Palette (Figure 3-65):
  a. Expand the Palette.
  b. From the list of items, select the one you want to add and drag it to the desired position within the dashboard. Repeat this step until you have added all desired items.

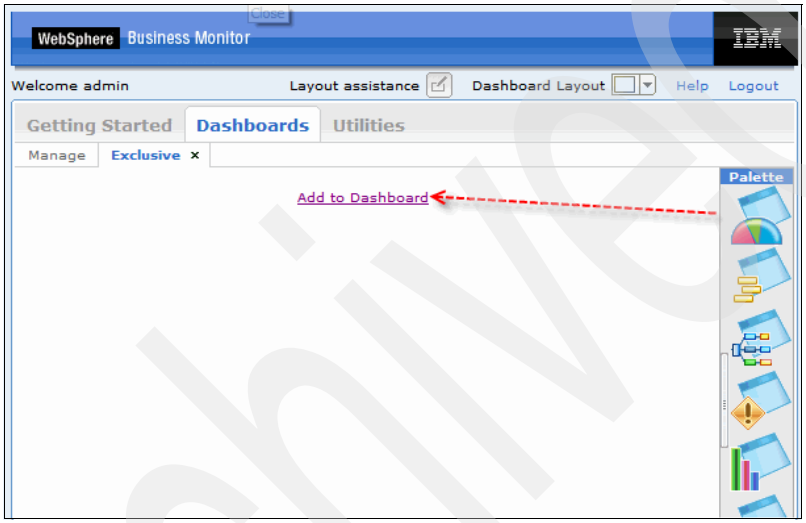


Figure 3-65 Adding data from a dashboard by using the Palette

Archived

- From a dashboard, if the dashboard layout outline is enabled (Figure 3-66):
  a. Click **Add to Dashboard**.
  b. From the list of available items, select the one that you want to add to the dashboard. To save your selection to the dashboard, click **OK**. To exit without saving, click **Cancel**.

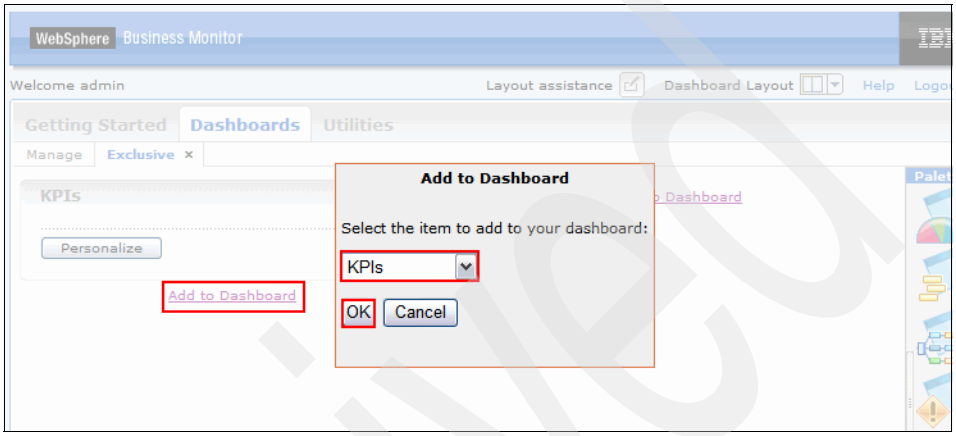


Figure 3-66 Adding data from a dashboard, if the dashboard layout outline is enabled

The selected item is added to your dashboard. You can then move the item to a different position on the dashboard by dragging it from the item's title bar to the desired location.

Personalizing KPIs on the dashboard

You can also use Personalize mode to configure the items on your dashboard as we explain in this section.

More information: The details of Alerts, Diagrams, Human Tasks, Instances, Dimension, Report and Alert Subscription are provided in the WebSphere Business Monitor V6.1 Information Center at the following Web address:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp`

Archive

To personalize KPIs on the dashboard:

1. On the dashboard, choose one of the following options to access the Personalize mode (Figure 3-67):
   - In the KPIs title bar, click the arrow to display the available options. Select **Personalize**.
   - Under KPIs, click **Personalize**.

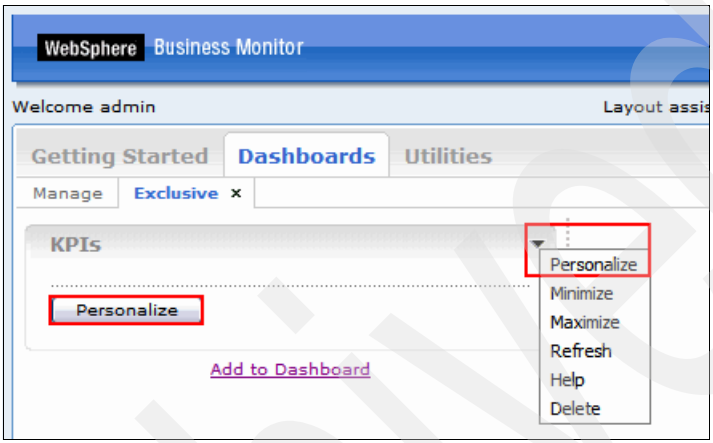


Figure 3-67 Personalizing KPIs

Archived

2. From the **Select KPIs** page (Figure 3-68):
   a. For Model version, select either All Versions or Latest Versions.
   b. Select the KPIs that you want to display in View mode. Click the model name to select all KPIs for the model.

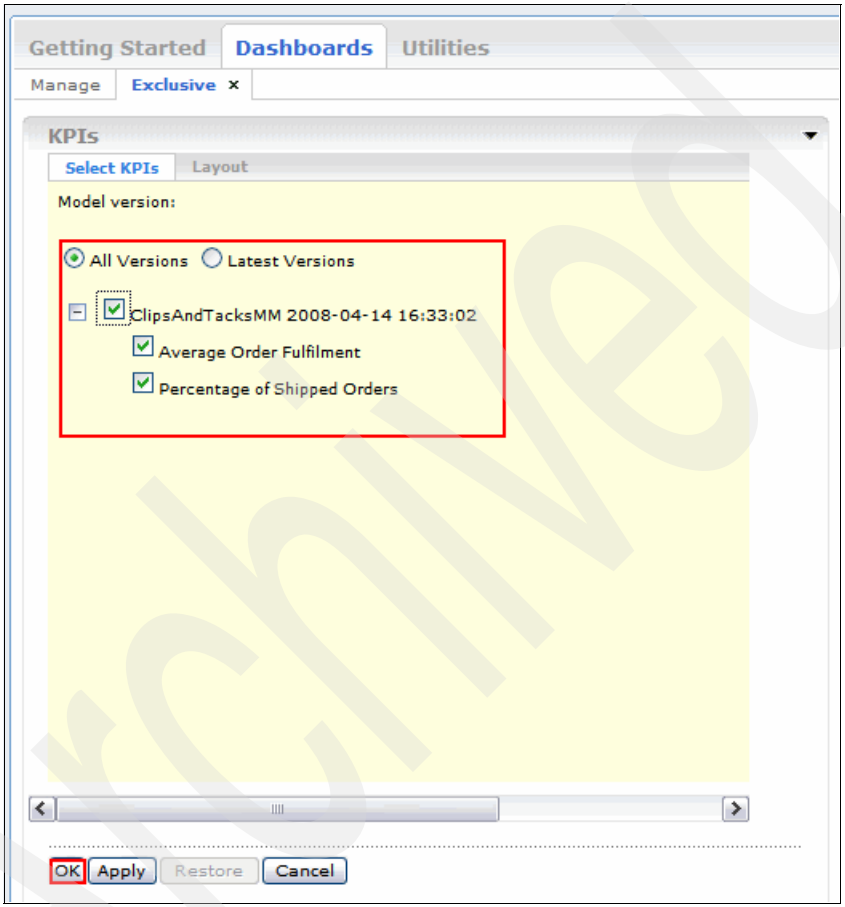


Figure 3-68 Select KPIs

3. To personalize the KPIs layout, click **Layout** from the Personalize mode toolbar (Figure 3-69 on page 124).
   a. Select one of the following display modes:
      - Table View
      - Half Gauge
      - Full Gauge
      - Bar Graph

Archived

b. Select one of the following layouts:

- Horizontal
- Vertical
- Grid

c. In the Columns field, if available, type the number of columns you want your KPIs to be displayed in.

d. Select one of the following sizes for the visualization elements:

- Small
- Medium
- Large

e. Type the number of seconds for the auto refresh rate of View mode.

f. Click **OK**.

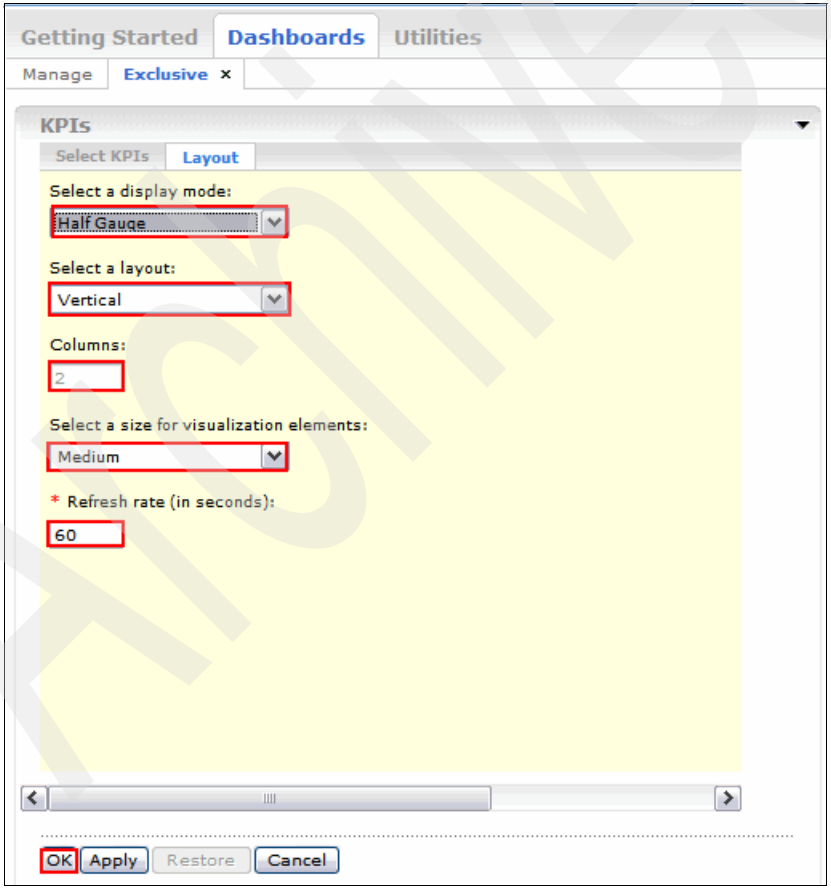


Figure 3-69 KPIs layout

Archived

The KPIs view is displayed as shown in Figure 3-70.

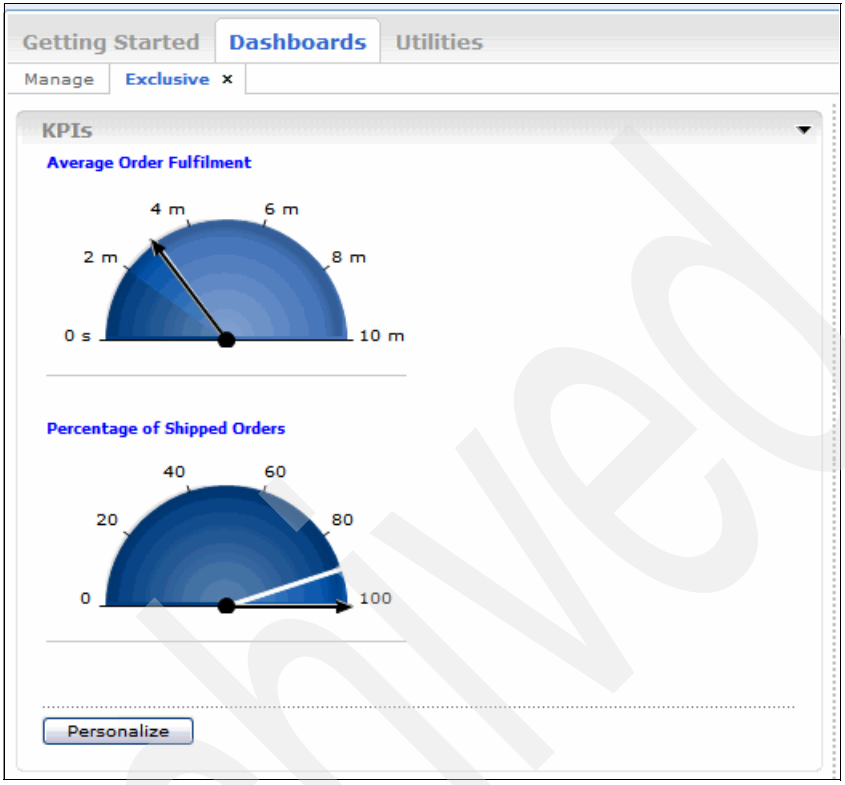


Figure 3-70 KPIs view

Archived

4. To view the KPI contextual menu from View mode, click a table view, half gauge, full gauge, or a bar graph. To open and edit the KPI properties, click **Go to properties** (Figure 3-71).

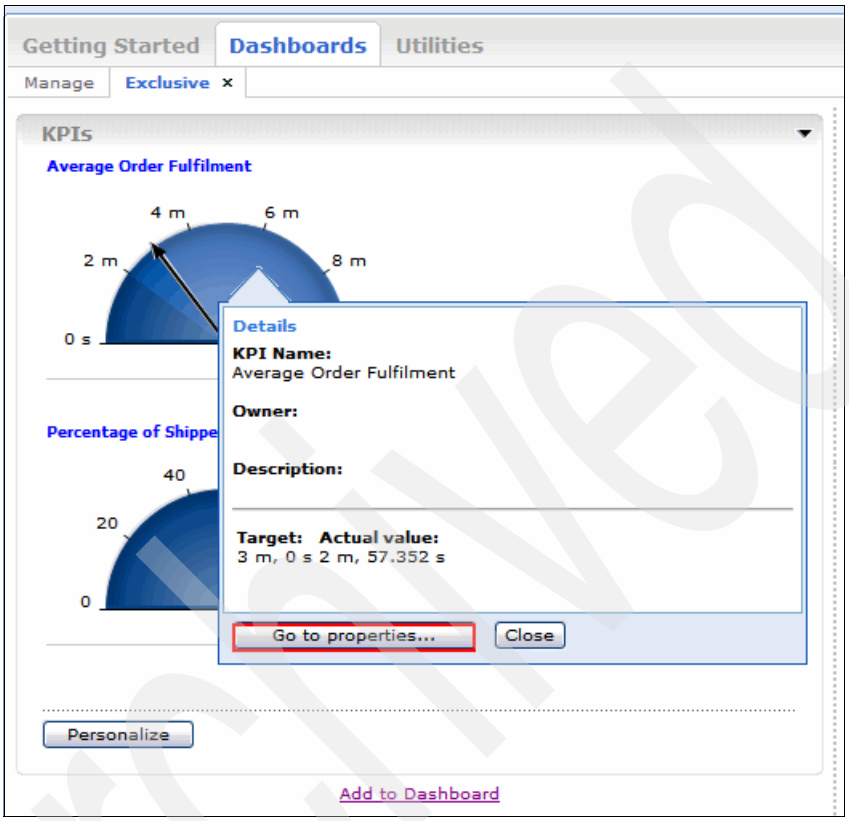


Figure 3-71 Go to properties

Archived

5. In the KPIs Properties panel (Figure 3-72), change the KPIs properties, such as ranges, colors, and so on, as needed.

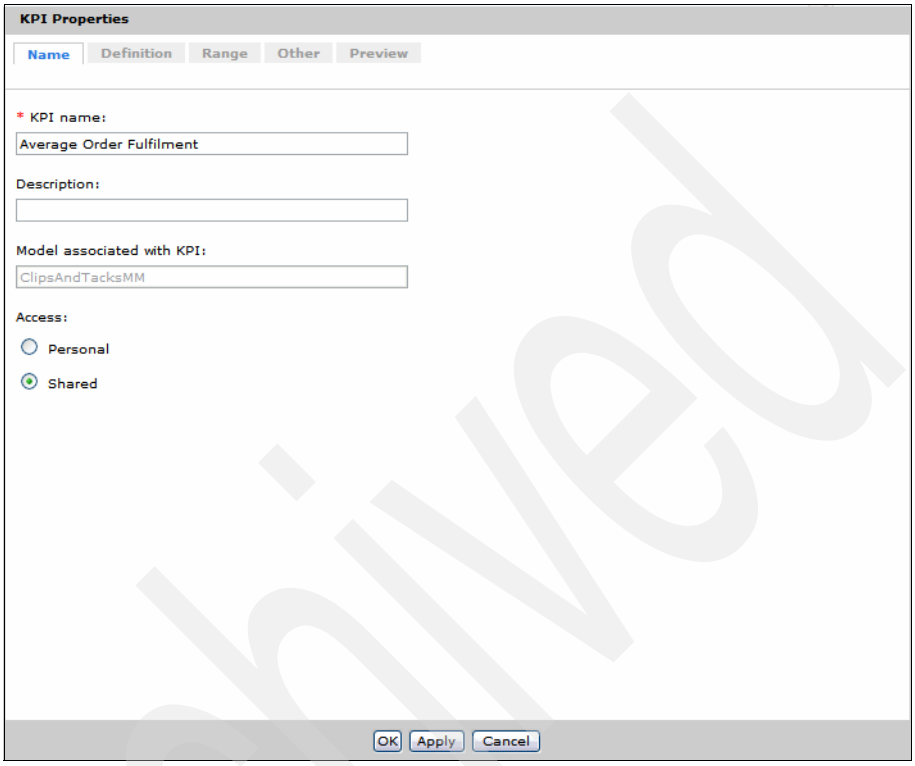


Figure 3-72 KPI Properties panel

Archived

6. After you adjust the KPI properties, click the **Preview** tab (Figure 3-73) to verify the changes and click **OK** to confirm them.

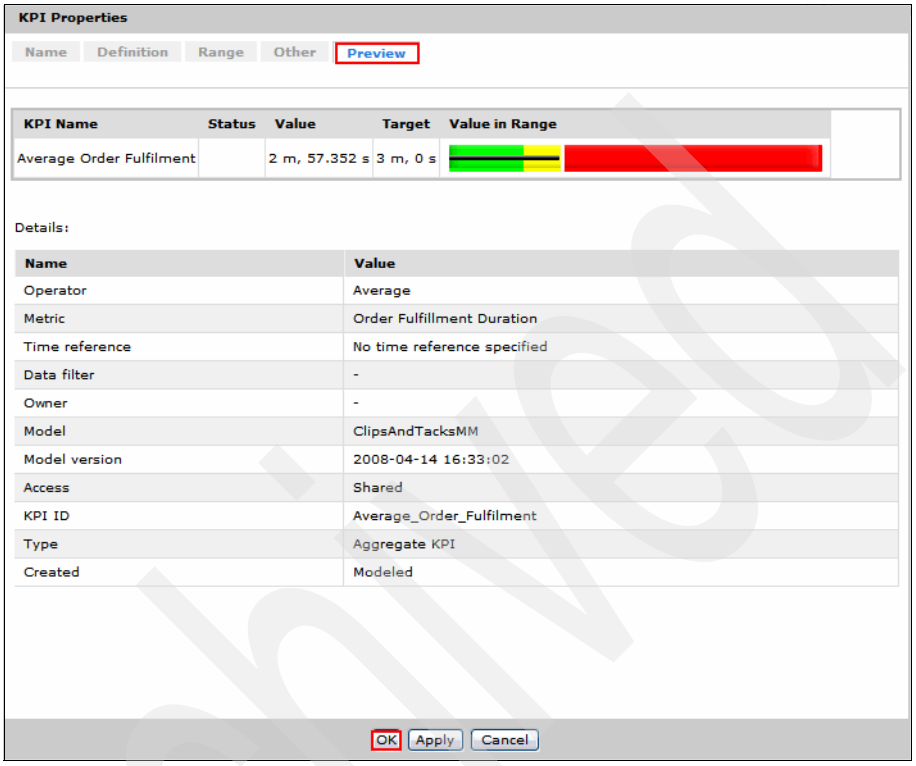


Figure 3-73 KPI Properties Preview tab

3.5.3 WebSphere Business Monitor KPI manager

WebSphere Business Monitor V6.1 provides the KPI manager so that business users can manage KPIs by using the dashboard, without any IT involvement. In earlier versions, KPIs were created and changed by the Model Editor exclusively and monitor models were redeployed for each KPI change. The KPI manager provides an overview of, and management capabilities for, all KPIs directly in the dashboard.

KPIs that are defined in a monitor model are called *modeled KPIs*. By using a dashboard, you can rename and modify the target and range values of the modeled KPIs. You can furthermore define new KPIs, which are called *dashboard KPIs*. A dashboard KPI can be a personal KPI that is visible solely to the user who created it, or it can be a public KPI that is visible to all users.

The authorization to create and modify KPIs is based on the users role. Roles are defined on a monitor model basis in the administrative console of the WebSphere Business Monitor server. There are three roles in the context of KPI management:

- Personal-KPI-Administrator

  With this role, users have authority to create non-shared (personal) KPIs. Only the owner and KPI-Administrator can view and update the created KPIs. Users of this role have read access to all metadata associated with the model, except KPIs.

- Public-KPI-Administrator

  In this role, users have the authority to create a shared (public) or non-shared (personal) KPIs. Other users can use and view shared (public) KPIs. Only the owner or KPI-Administrator can make changes to shared (public) KPIs. Users of this role have read access to all metadata associated with the model except KPIs.

- KPI-Administrator

  With this role, users have all the authority associated with KPI administration. Users of this role can create both shared (public) and non-shared (personal) KPIs. In addition, the KPI-Administrator can change the ownership of any KPIs. This role can view all KPIs, shared and non-shared. Users of this role can also update non-core properties and have read access to all metadata associated with the model except KPIs.

Dashboard KPIs are automatically merged into each new monitor model version deployed on the WebSphere Business Monitor server.

Archived

Adding a user to a WebSphere Business Monitor role

If WebSphere Business Monitor has security turned on, the following steps are required:

1. In the WebSphere Business Monitor administrative console (Figure 3-74), select **Security** → **Monitor Data Security**. In the right pane, click the **root** link.

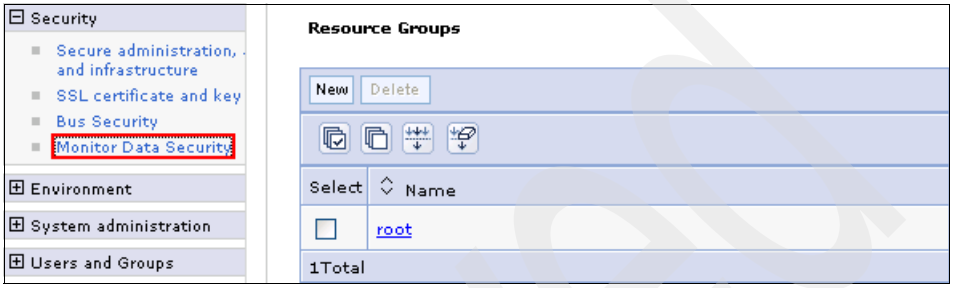


Figure 3-74 Navigating to Monitor Data Security

2. Select the **KPI-Administrator** radio button and click the **Users** button (Figure 3-75).

Figure 3-75 Adding a user to the KPI-Administrator role

Archived

3. Click the **Search** button. Select the wsadmin user and click the **>** button to move the user to the *Selected* box (Figure 3-76).

   Note: The available user may be different, depending on the underlying user registry.

   Click **OK**.

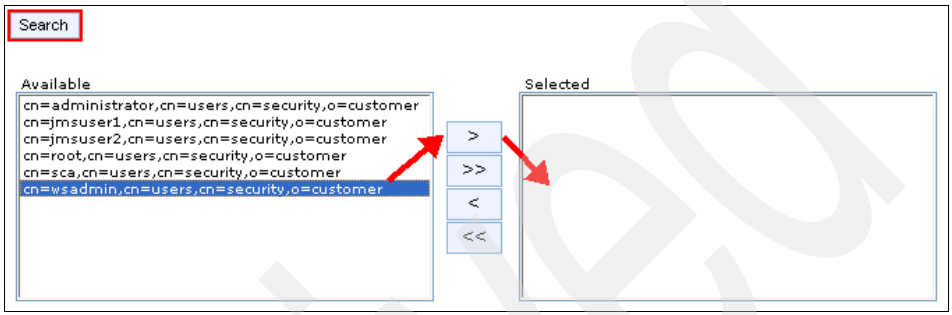


Figure 3-76 Configuring the KPI-Administrator

4. Restart the WebSphere Business Monitor server.

Viewing the KPI Manager actions

To view the KPI Manage actions:

1. Log in to the WebSphere Business Monitor Business dashboard by using a user with a KPI-Administrator role.

   Note: You can access the Business dashboard of WebSphere Business Monitor from the following URL:

   `http://<host name>:<WC_defaulthost port>/BusinessDashboard`

Archived

2. Switch to the **Utilities** tab (Figure 3-77). On the **KPI Manager** tab, select a monitor model.

   A list of all defined KPIs is shown. The created column indicates if a KPI is defined in the monitor (Modeled) or in the Dashboard (Dashboard). The Access column states if the KPI is private (Personal) or public (Shared).

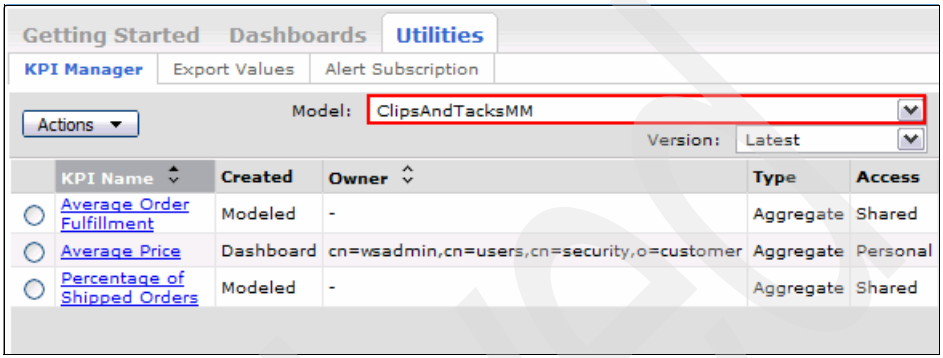


Figure 3-77 KPI Manager tab

3. Click the **Actions** button to view the available actions (Figure 3-78).

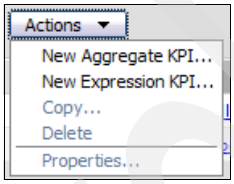


Figure 3-78 KPI manager Actions

Modifying a modeled KPI

To modify a modeled KPI:

1. On the **KPI Manager** tab, select a modeled KPI and click the **Actions** button.
2. Select **Properties**.

Archive

3. Click the **Name** tab (Figure 3-79). Modify the name, description, and access if necessary.

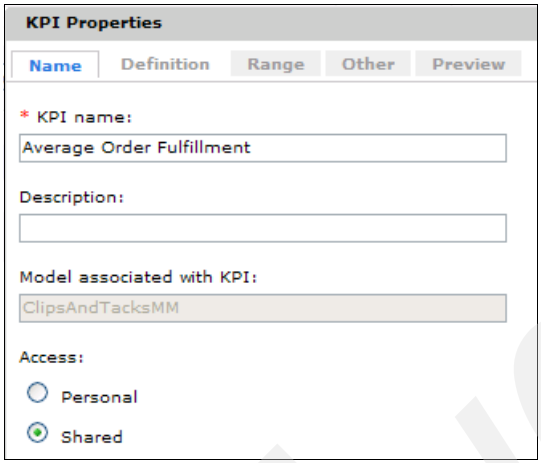


Figure 3-79 KPI Properties Name tab

4. Click the **Range** tab (Figure 3-80). Modify the target value and range names, values, and colors if necessary.

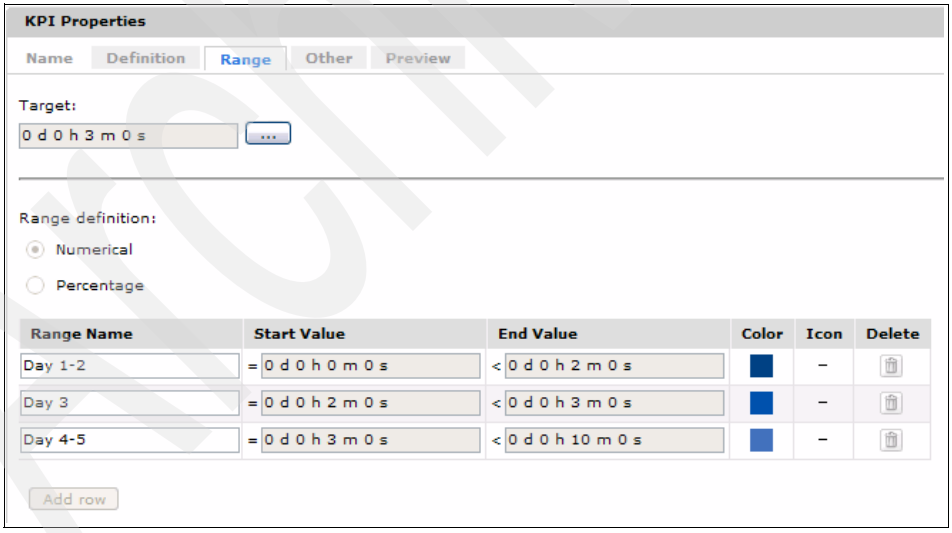


Figure 3-80 KPI Properties Range tab

Archived

5. Click the **Other** tab (Figure 3-81). This page is for the KPI Administrator to set and modify the KPI owner.

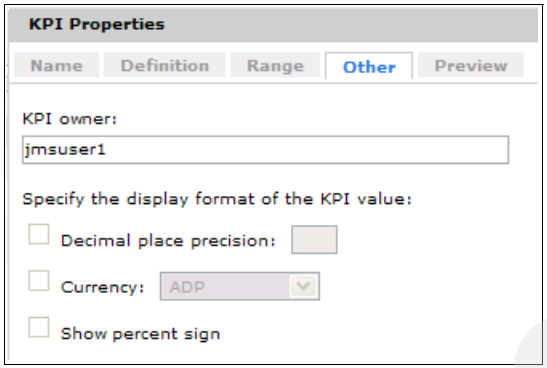


Figure 3-81 KPI Properties Other tab

6. Click the **Preview** tab (Figure 3-82) to see a preview of the configured changes.

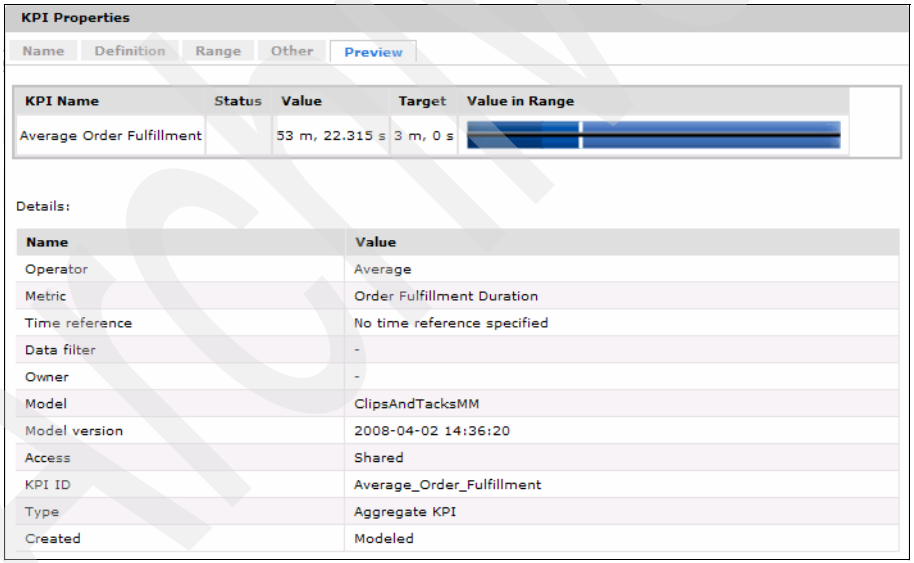


Figure 3-82 KPI Properties Preview tab

7. Click **OK** to save the changes.

Archived

Creating a new Aggregate KPI

An aggregate KPI is based on metric defined in the monitor model. To create a new Aggregate KPI:

1. Click the **Actions** button.
2. Select **New Aggregate KPI**.
3. Enter the KPI name, Definition, Range, and Other values.
4. Click **OK** to save the settings.

For more information about creation of an Aggregate KPI, refer to “Creating an aggregate KPI” in the WebSphere Business Monitor information center:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.btools.help.monitor61.doc/dash/kpiexplorer_aggregate_web.html

Creating a new Expression KPI

An expression KPI is based an expression value. To create a new Expression KPI:

1. Click the **Actions** button.
2. Select **New Expression KPI**.
3. Enter the KPI name, Definition, Range, and Other values.
4. Click **OK** to save the settings.

For more information about the creation of an Expression KPI, refer to “Creating an expression KPI” in the WebSphere Business Monitor information center:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.btools.help.monitor61.doc/dash/kpiexplorer_expression_web.html

Copying a KPI

To copy a KPI:

1. Select a KPI in the KPI list and click the **Actions** button.
2. Select **Copy**.
3. In the Properties panel of the new KPI, you see pre-filled with values of the base KPI. Change KPI name, range, definition, and other values. Then click **OK** to save the settings.

Deleting a KPI

To delete a KPI:

1. Select a KPI in the KPI list and click the **Actions** button.
2. Select **Delete** to remove the KPI.

Archived

3.6 Summary

In this chapter, we provided an overview of how to install WebSphere Business Monitor and WebSphere Business Monitor Toolkit. We explained basic usage of WebSphere Business Monitor including how to deploy, update, version, and remove the monitor model; create a dashboard; and use the KPI Manager.

Archived

Part 2

Business activity monitoring

In this part, we provide step-by-step examples on how to implement business activity monitoring (BAM) by using WebSphere Business Monitor V6.1 and various IBM products:

- Chapter 4, “Business scenario: ClipsAndTacks” on page 139

  This chapter introduces the ClipsAndTacks Office Supplies Pte. Ltd. scenario and describes how the scenario relates to the remaining chapters in this book.

- Chapter 5, “WebSphere Process Server” on page 149

  This chapter demonstrates the integration of WebSphere Business Monitor with WebSphere Process Server to implement a BAM solution.

- Chapter 6, “WebSphere ESB and IBM Adapters” on page 249

  This chapter demonstrates the integration of WebSphere Business Monitor with WebSphere Enterprise Service Bus and WebSphere Adapters to implement a BAM solution.

© Copyright IBM Corp. 2008. All rights reserved.

- Chapter 7, “FileNet P8 BPM” on page 351

  This chapter demonstrates the integration of WebSphere Business Monitor with FileNet P8 BPM to implement a BAM solution.

- Chapter 8, “WebSphere MQ Workflow” on page 481

  This chapter demonstrates the integration of WebSphere Business Monitor with WebSphere MQ Workflow to implement a BAM solution.

- Chapter 9, “Custom event emitter” on page 599

  This chapter demonstrates how to build a custom Java application to emit business events to WebSphere Business Monitor.

- Chapter 10, “WebSphere MQ” on page 665

  This chapter demonstrates the integration of WebSphere Business Monitor with WebSphere MQ to implement a BAM solution.

Archived

4

Business scenario: ClipsAndTacks

In this chapter, we describe the background business scenario on which the subsequent monitoring examples are based. This scenario is based loosely on a case study description found in *Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products*, SG24-7148.

In addition, we describe several business activity monitoring (BAM) needs and how they can be addressed using WebSphere Business Monitor. We discuss the business processes, the applications involved, and the business process management products used.

© Copyright IBM Corp. 2008. All rights reserved.

4.1 Case study: ClipsAndTacks Office Supplies Pte. Ltd.

This case study describes a fictional company, ClipsAndTacks Office Supplies Pte. Ltd. (shortened to *ClipsAndTacks* in this book), that faces many internal and external challenges. Some challenges are business related and some are systems related. We discuss the need for ClipsAndTacks to measure its business performance and monitor its business processes. We mention various scenarios that we explore later in this book.

4.1.1 Background

ClipsAndTacks is a medium-sized office supply company operating in eastern Canada and the northeastern United States. The company has grown slowly and achieved a significant customer base through its excellent customer service practices and reputation for quality products.

In recent times, ClipsAndTacks has been growing and changing. Competitive pressures have demanded changes in their traditional business approach. Examples of the changes are easy to see. The company recently changed from handling orders strictly by phone to handling orders from the Web. Various departments have begun automating business processes by using business process management (BPM) tools. A new division was recently created by acquisition. As a result of these and other changes, the business is not as easily understood or manageable as it once was.

Management is feeling the pressure. Where once they were in charge of a self-contained business, built slowly and controlled directly, now they must manage a business that is growing and changing rapidly. They have adopted business process modeling and management in several areas to help keep those processes under control and to allow rapid response to changing needs. At the same time, many processes are controlled manually. In either case, meaningful visibility into the day-to-day operations is needed.

ClipsAndTacks management is seeking a way to deal rationally with the changing business as a whole.

4.1.2 Business challenges

ClipsAndTacks is a successful business. The company is both growing and adapting in a competitive marketplace. As the business expands, management is faced with the challenge of understanding its business performance and processes in a timely fashion. Traditional reporting at the end of the month does

not allow for the rapid response needed in today's marketplace. ClipsAndTacks management needs a way to monitor its business performance in near real time.

Management believes in the idea of BAM, measuring and controlling the business through a *dashboard* concept. They have worked with consultants to choose WebSphere Business Monitor as their enterprise BAM solution. Now they must make the concept a reality.

While their goal is to understand and control the business as a whole, it takes time and effort to identify appropriate key performance indicators (KPI), to identify from where the appropriate data may be drawn, and to establish meaningful dashboards for varying layers of management. Looking at the business, the starting points presented in the following sections are identified for BAM and are used as the examples in this book.

Order handling process

Within ClipsAndTacks, a new managed process exists for order handling. This process was created in response to customer feedback and implemented in the original company (pre-acquisition) using WebSphere Process Server. The process starts with the submission of an order via a Web interface. The steps are managed through shipping by automating a business model and drawing on various technology services, such as a business rules engine and human task management.

New division order handling process

After the recent acquisition, ClipsAndTacks order handling is carried out by two separate divisions. The recently acquired division uses WebSphere MQ Workflow to manage part of the business, while the pre-existing ClipsAndTacks division uses WebSphere Process Server to manage the same activities. In both cases, even though the process model is essentially the same, the two BPM engines continue to be used to avoid the cost of conversion in the short term. However, management demands visibility into both processes.

Contract management process

Within ClipsAndTacks, the contract management department uses FileNet P8 to store customer contracts and related documents. When challenged to adopt business process management, it turned to FileNet P8 BPM to manage its document centered processes. Management now requires visibility into the workings of the contract management process even though it uses a different BPM engine from those in order processing.

External shipping service

ClipsAndTacks uses an external, international shipping company, ITSO Logistics, to deliver its orders. In many ways, the performance of the shipping company reflects directly on the satisfaction of its customers. In the past, temporary slowdowns at the shipping company were only evident after the fact, in monthly reports from the ClipsAndTacks shipping department or reflected in customer complaints. Management wants visibility into the current performance of the shipping company.

Order processing database

While monitoring the managed processes dealing with order handling will give an indication of their performance, other departments within ClipsAndTacks, such as vendor management, deal with orders as well. For an overall view of how orders are handled, it is deemed necessary to mine the common order database for related business events. When orders are updated in the database, business events are generated and sent to WebSphere Business Monitor for processing.

Customer credit check service

An existing customer credit check service is used by the managed processes as well as other unmanaged processes. Handling orders and other requests in a timely fashion demands that the credit check service respond efficiently. It is identified as a point from which business events can be generated and sent to WebSphere Business Monitor.

Summary of problems

ClipsAndTacks management has identified the following problems to be addressed in its push to adopt a BAM solution:

- Must measure business performance at many touch points and display those indications through a common mechanism (that is, a dashboard)
- Must extract business events from managed and nonmanaged processes, from varying types of technology
- Must monitor business processes in meaningful detail:
  - Processes managed by WebSphere Process Server
  - Processes managed by WebSphere MQ Workflow
  - Processes managed by FileNet P8
  - Processes managed by people and embodied in existing application assets, such as databases and messaging

Archive

4.2 Order handling process

ClipsAndTacks has implemented an order submission process so that it can compete with office supply companies in its target market. This process was developed in response to a common complaint in customer surveys. The complaint was that the customers wanted to place an order online without having to interface with customer service unless they had a question or specific need. The new process eliminated the need for contact between customers and customer service representatives when an order is placed.

Customers can now browse the ClipsAndTacks product catalog online or in printed copy and enter their own order information using a Web application. Customers who have a customer number can enter it, 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, is available 24 hours a day, 7 days a week.

Due to the recent acquisition, ClipsAndTacks has two divisions that deal with orders, but use different BPM engines. For both, in terms of steps and integration touch points, the process is exactly the same.

When the customer submits the order, a business rules engine is checked to ensure appropriate action on the order. Orders of gold customers are reviewed if the total price is higher than $1750. Silver customer orders are reviewed if the total price is higher than $1250. All other customer orders must be reviewed for a total price higher than $750. 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. Based on the review, the order manager decides whether to send the order for fulfillment or to cancel the order and notify the customer.

This process include the following highlights:

- Customers can order online (shortens the order process).
- Orders are accepted 24 hours a day, 7 days a week.
- Implementation of a rules or policy engine is now included.
- Certain customer orders can be pre-approved by rules. Only remaining orders must be approved by the order manager. This shortens the average order time and increases the percentage of approved orders.

Archive

Figure 4-1 shows a representation of the business process from WebSphere Business Modeler.

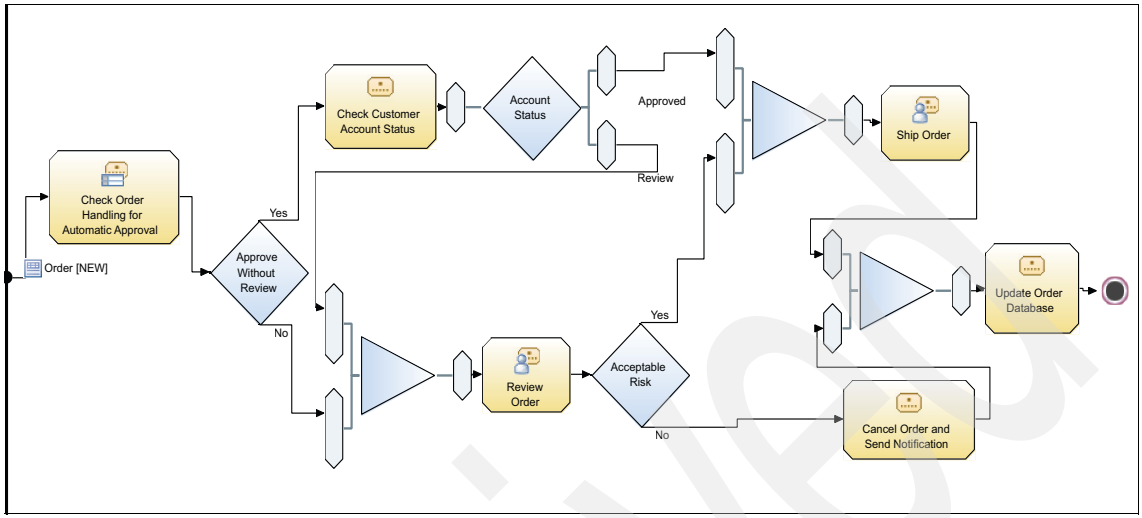


Figure 4-1 Workflow model of the order handling process

This process is initiated by an incoming order. It moves through different check points before it is shipped or cancelled. These check point are business rules or human tasks. ClipsAndTacks is interested in an overview of the running process instances and in KPIs, such as average order fulfillment duration or percentage of shipped order. Furthermore, they want an idea about how many orders are shipped in total and the ability to drill down to totals by countries and cities. Another requirement is to provide process and KPI information displayed in a given graphical representation of the order handling process flow.

Order handling process: This order handling process is the basis for the following chapters:

- In Chapter 5, “WebSphere Process Server” on page 149, we show how you can use WebSphere Business Monitor to monitor this process by executing it within WebSphere Process Server.
- In Chapter 8, “WebSphere MQ Workflow” on page 481, we show how you can use WebSphere Business Monitor to monitor this process by executing it within WebSphere MQ Workflow.

4.3 Contract management process

ClipsAndTacks deals with a large number of domestic and international business customers. Each customer contract has an amount, and sometimes large amounts, of related content data including documents, images, audio files, and video files. This content must be stored safely and managed effectively. The contract management department uses FileNet P8 as its content repository and FileNet P8 BPM to manage its document centric workflow.

ClipsAndTacks contract management has the following business requirements to serve:

- All structured and unstructured content related to contract management must be stored and managed.
- Contract management business processes must be defined, versioned, analyzed, simulated, tracked, and managed.
- The status of contract management processes and the overall performance of those processes must be visible to management in near real time.

Within the contract management department at ClipsAndTacks, indexers work with business customers to establish contractual relationships. They collect and analyze contract information and then use FileNet P8 to create contract cases and store related documents. When assembled, each contract case is submitted for subsequent processing.

While working with a customer, indexers can suspend the contract cases pending further additions or modifications. After the cases are complete, the cases are submitted for review.

A reviewer sees a list of contract cases submitted by indexers. By using the FileNet P8 system, a reviewer can examine a contract case, its attached documents, and its history and audit information. After checking these documents and adding annotations, the reviewer chooses an approver who is assigned the case for final approval. If further processing is needed, the reviewer can suspend the contract for later processing. Alternatively, if there is sufficient reason, the reviewer can reject the contract directly.

An approver sees all contract cases that are sent for approval. The approver selects from a list of one or multiple (known as *bulk processing*) contracts and then can approve or reject the contract.

Archived

The process can be summarized as follows:

1. Collect contract information.
2. Create and index the corresponding case.
3. Review and evaluate the contract case.
4. Approve or reject the contract case.

Several roles are involved in this process flow:

- Indexer

  This person works with the customer to collect contract information and then creates and indexes a contract case.

- Reviewer

  This person reviews the contract case and then sends it to an approver or rejects it back to an indexer for further processing.

- Approver

  This person approves or rejects a previously reviewed contract case.

- Supervisor

  This person can assume the role of an indexer or a reviewer and can search.

Figure 4-2 shows an overview of the contract management process.

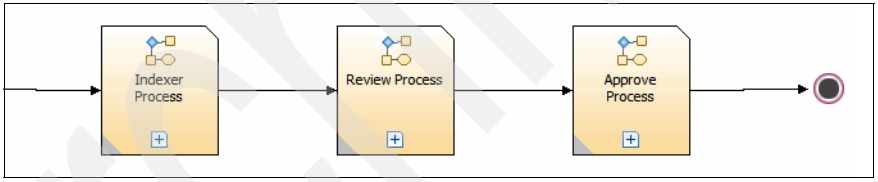


Figure 4-2 High-level contract management process

Note: In Chapter 7, “FileNet P8 BPM” on page 351, we show how you can use WebSphere Business Monitor to monitor this process by executing it in FileNet P8 BPM.

4.4 Order database

ClipsAndTacks has an order database, which a common repository used by all applications and areas that deal with the creation or fulfillment of orders. This database contains the status of an order and all its details. Entries in the database tables are created when an order is first placed, updated as necessary

during the processing of the order, and updated to reflect when an order has been shipped. This database reflects the true state of all orders within the company from the companies inception.

More information: The idea of monitoring a centrally used database for business events is demonstrated in Chapter 6, “WebSphere ESB and IBM Adapters” on page 249.

4.5 Shipping

ClipsAndTacks uses a company called ITSO Logistics as its primary shipping provider. ITSO Logistics is a major, although not well known, international shipping company. It ships packages for ClipsAndTacks to over 100 countries worldwide.

As part of their arrangement, ClipsAndTacks informs ITSO Logistics when a package or group of packages is ready to be picked up for shipment. Since this arrangement has been in place for many years, the initial form of communication between the companies, which remains in place to this day, is via FTP through a private trading network. Upon receipt of the shipment-ready file, ITSO Logistics schedules a truck (if one with capacity is not already in transit) to pick up the packages at the ClipsAndTacks shipping dock. The truck transports the packages to the ITSO Logistics distribution center for routing and any appropriate customs’ tasks. From the distribution center, ITSO Logistics sends the packages on their way by various means at its disposal.

ITSO Logistics internally tracks each step in the process but ultimately reports back to ClipsAndTacks a shipment notification that contains a summary for each individual package shipped. The summary includes the total time, to the day, that it took for a delivery. This notification is delivered as a file via FTP. Figure 4-3 illustrates the basic flow.

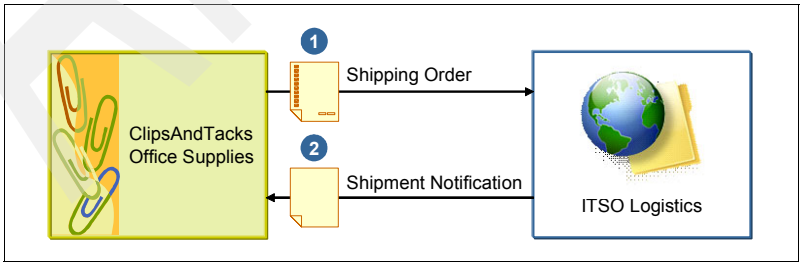


Figure 4-3 Interface to the external shipper

While ITSO Logistics is a long-term business partner, ClipsAndTacks wants to assess its partner's ongoing performance. Monitoring of the shipments allows them to anticipate customer situations and to better evaluate if or when a change of shipping vendor is needed.

More information: Monitoring an external entity by using events generated at the interface is demonstrated in Chapter 9, "Custom event emitter" on page 599.

4.6 Credit check

The credit check service is a mainframe application that acts as an automated reporter on the status of a customer's credit. The details of how a credit decision is made are private to the financial arm of the business. The application is written in COBOL and uses WebSphere MQ as a transport for requests and replies. The format of a request is a proprietary format based on COBOL copybooks.

While business processes are involved in gathering information, maintaining that information, and responding to requests, the processes are not managed in any way by a BPM engine. In fact, this particular business function has been available since the early days of ClipsAndTacks, even before the widespread use of computer technology in handling order processing.

The credit check service, provided as a mainframe application, is used by many areas of the ClipsAndTacks enterprise. As such, it is a key function in the overall business operations and must be monitored.

More information: Monitoring an existing application with ad hoc events and transporting them to the CEI is demonstrated in Chapter 9, "Custom event emitter" on page 599.

4.7 Summary

In this chapter, we introduced the ClipsAndTacks company. We described several areas that can benefit from BAM technology. We used their order handling and contract management business processes to demonstrate how the WebSphere Business Monitor can be used in conjunction with three different BPM engines to monitor the ClipsAndTacks business. In addition, we show how their other applications, using such technologies as DB2 and WebSphere MQ, can be monitored as part of an overall BAM solution.

5

WebSphere Process Server

In this chapter, we discuss the integration of WebSphere Process Server V6.1 into WebSphere Business Monitor V6.1.1. Based on the ClipsAndTacks Order Handling process introduced in Chapter 4, “Business scenario: ClipsAndTacks” on page 139, we explain how to generate and extend a monitor model by using the WebSphere Business Monitor Toolkit. We also describe the deployment of the business process to a WebSphere Process Server and deployment of the monitor model to a WebSphere Business Monitor server. In addition, we provide information about a WebSphere Business Monitor dashboard set up.

© Copyright IBM Corp. 2008. All rights reserved.

5.1 Introduction to WebSphere Process Server

IBM WebSphere Process Server is a business integration server. It is built to support solutions based on service-oriented architecture (SOA). It can be used to build advanced business processes and traditional business integration. WebSphere Process Server is based on WebSphere Application Server and combines the best features of WebSphere MQ Workflow, WebSphere InterChange Server, and WebSphere Business Integration Server Foundation.

Build 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.

As shown in Figure 5-1, WebSphere Process Server includes three layers:

- SOA core
- Supporting services
- Service components

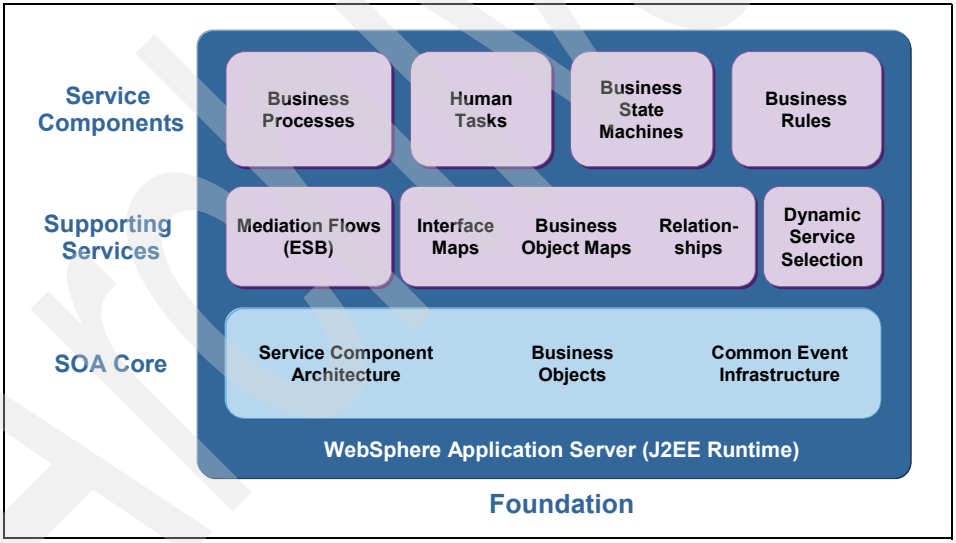


Figure 5-1 WebSphere Process Server architecture

5.1.1 SOA core

The SOA core consists of the Service Component Architecture (SCA), business objects, and the Common Event Infrastructure (CEI). SCA and business objects provide uniform invocation and data representation programming models for

applications running on WebSphere Process Server. CEI enables monitoring and management capabilities for these applications.

5.1.2 Supporting services

On top of the runtime infrastructure and SOA core, WebSphere Process Server offers a variety of service components that address several transformation challenges for connecting components and external artifacts. They are needed in any integration solution, including data transformation and synchronization services. Supporting services include the following service components:

- Mediation flows

  Mediation flows intercept and modify messages that are passed between services providers and service requesters.

- Interface maps

  Interface maps reconcile differences between components that have different interfaces.

- Business object maps

  Business object maps support mappings between business objects.

- Relationships

  Relationships are supporting services in WebSphere Process Server applications that establish an association between data of several data types.

- Selectors

  A selector can be used for dynamic selection and invocation of different services that all share the same interface.

5.1.3 Service components

All integration artifacts running on WebSphere Process Server are represented as components with well defined interfaces. WebSphere Process Server provides the following service components:

- Business processes

  A business process component in WebSphere Process Server implements a Business Process Execution Language (BPEL) compliant process.

- Human tasks

  Human tasks are stand-alone service components that can be used to assign work to employees.

Archived

- Business state machines

  Business state machines are service components that specify a sequence of states, responses, and actions an object can have during its life cycle.

- Business rules

  With business rules, you can implement and enforce a business policy through an externalizing business function. This enables dynamic changes of a business process for a more responsive business environment.

For more information about WebSphere Process Server, refer to the following Web address:

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

5.2 Overview of the integration of WebSphere Process Server into WebSphere Business Monitor

WebSphere Process Server provides a tight integration point into WebSphere Business Monitor. CEI is part of the WebSphere Process Server SOA core layer. It allows service components as BPEL processes to emit events that can be captured by WebSphere Business Monitor for real-time monitoring of business processes.

Figure 5-2 on page 153 illustrates the steps to monitor the business processes of WebSphere Process Server. These steps are explained as follows:

1. Enable event emitting for service components in WebSphere Integration Developer. Each service component provides a set of predefined events that can be emitted to CEI.
2. Deploy the WebSphere Process Server application to a WebSphere Process Server.
3. Generate a monitor model based on a service component by using the WebSphere Business Monitor Toolkit.
4. (Optional) Extend the monitor model with monitoring components such as metrics, alerts, key performance indicators (KPIs), or dimensions.
5. Generate the J2EE projects based on the monitor model in WebSphere Integration Developer.
6. Deploy the generated J2EE projects on a WebSphere Business Monitor server.

7. In case the WebSphere Process Server application and the monitor model are not deployed on the same server, configure the CEI of the WebSphere Process Servers to act as a remote CEI.
8. Develop the WebSphere Business Monitor dashboard.
9. Generate events by execution the WebSphere Process Server application and monitor them by using a dashboard.

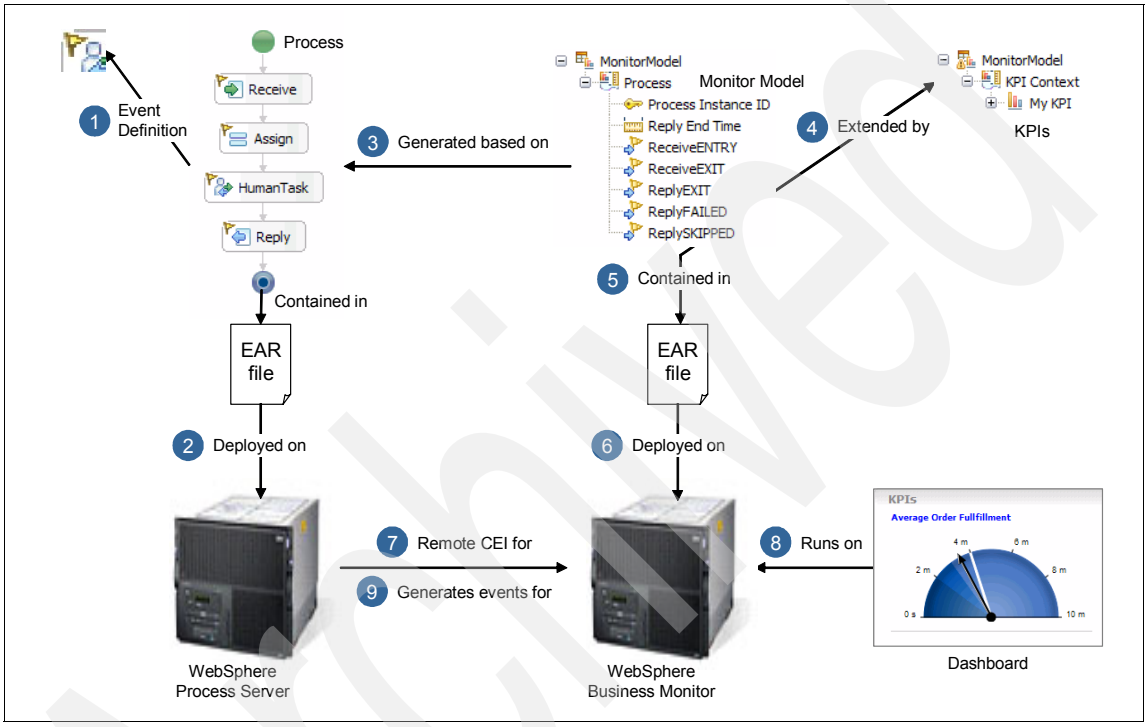


Figure 5-2 Steps to monitor a WebSphere Process Server process

5.3 Integrating WebSphere Process Server into WebSphere Business Monitor

In this section, we provide an example of integrating WebSphere Process Server into WebSphere Business Monitor. As described in Chapter 4, “Business scenario: ClipsAndTacks” on page 139, one of ClipsAndTacks’ departments uses WebSphere Process Server to execute the Order Handling process. This process is initiated by an incoming order that goes through different check points before it is either shipped or cancelled. These check points are business rules or human tasks. ClipsAndTacks is interested in an overview of running process

instances and KPIs such as average order fulfillment duration and percentage of shipped order. They furthermore want an idea about how many orders are shipped drilled down to countries and cities. Another requirement is to provide process and KPI information displayed in a given graphical representation of the Order Handling process flow.

We explain the BPEL implementation of the Order Handling process and provide a step-by-step guide to monitor that process in a WebSphere Business Monitor dashboard. ClipsAndTacks is running a WebSphere Process Server and a WebSphere Business Monitor on two machines as unsecured stand-alone servers that require a remote CEI configuration.

5.3.1 Software requirements

To run the sample, the following software is required:

- WebSphere Integration Developer V6.1.003
- WebSphere Process Server V6.1

  This scenario uses the integrated test environment server profile provided by WebSphere Integration Developer. Security is disabled.
- WebSphere Business Monitor V6.1.1 – Toolkit Installation, including the Monitor Development Environment
- WebSphere Business Monitor V6.1.1

  Security is disabled.

5.3.2 Importing the ClipsAndTacks BPEL process

ClipsAndTacks uses an order handling process BPEL implementation. We explain how to import the process into WebSphere Integration Developer and describe all process components.

At this point, we assume that you are familiar with the WebSphere Integration Developer 6.1 process development. If not, refer to *Getting Started with WebSphere Process Server and WebSphere Enterprise Service Bus Part 1: Development*, SG24-7608.

Archived

To import the process into WebSphere Integration Developer:

1. In WebSphere Integration Developer, set up the Business Integration environment:
   a. Launch WebSphere Integration Developer.
   b. During the startup, when you see the Workspace Launcher window (Figure 5-3), in the Workspace field, select a new workspace and click **OK**.

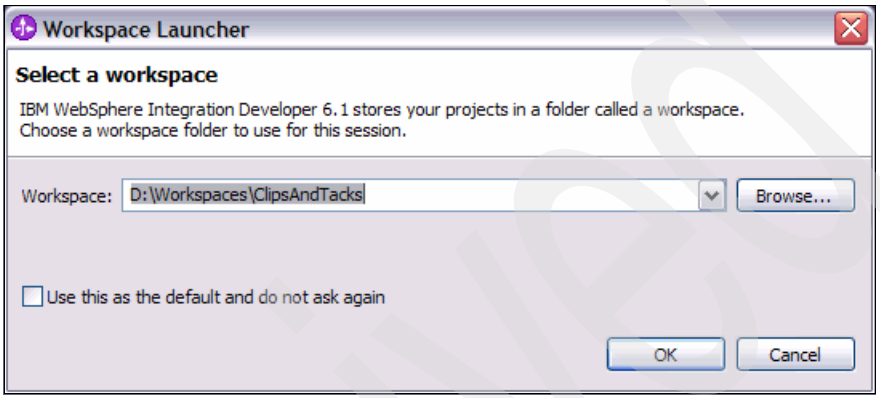


Figure 5-3 Creating a ClipsAndTacks workspace

   c. Open the Business Integration perspective by selecting **Window** → **Open Perspective** → **Business Integration**.

Archived

2. Import the Project Interchange file:
   a. In WebSphere Integration Developer, select **File** → **Import**.
   b. In the Import window (Figure 5-4), expand **Other** and select **Project Interchange**. Then click **Next**.

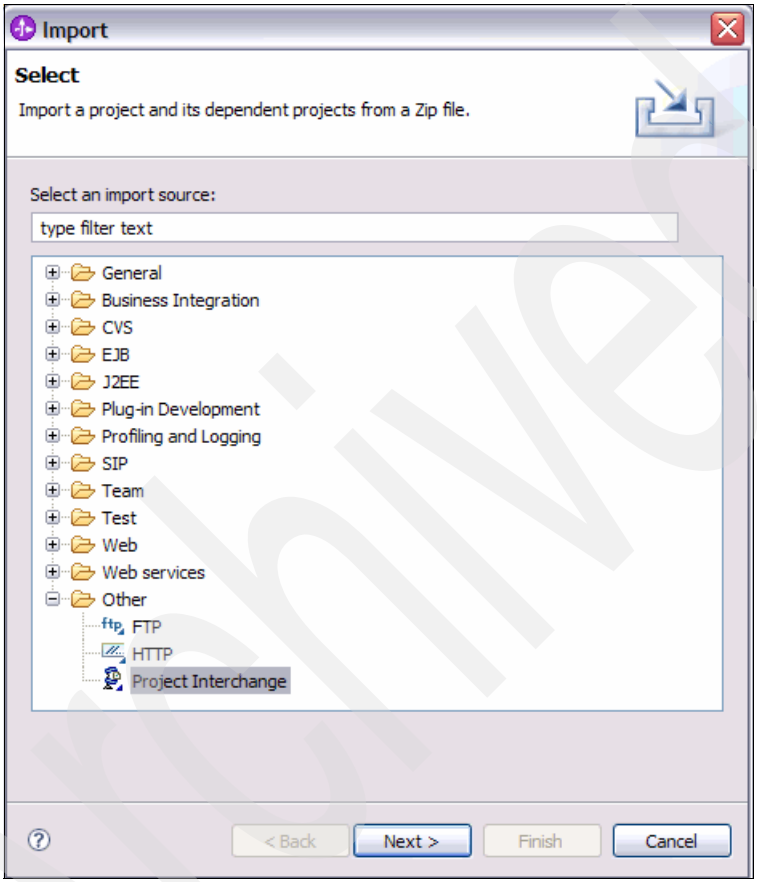


Figure 5-4 Importing a Project Interchange file

Archived

c. In the Import Project Interchange Contents window (Figure 5-5):

i. For the From zip file field, click the **Browse** button and select the **ClipsAndTacks_BPEL_Process.zip** file.

> **Note:** The ClipsAndTacks_BPEL_Process.zip file is provided in the additional material that is available for this book. See Appendix B, “Additional material” on page 801, for more information.

ii. Click the **Select All** button.

iii. Click **Finish**.

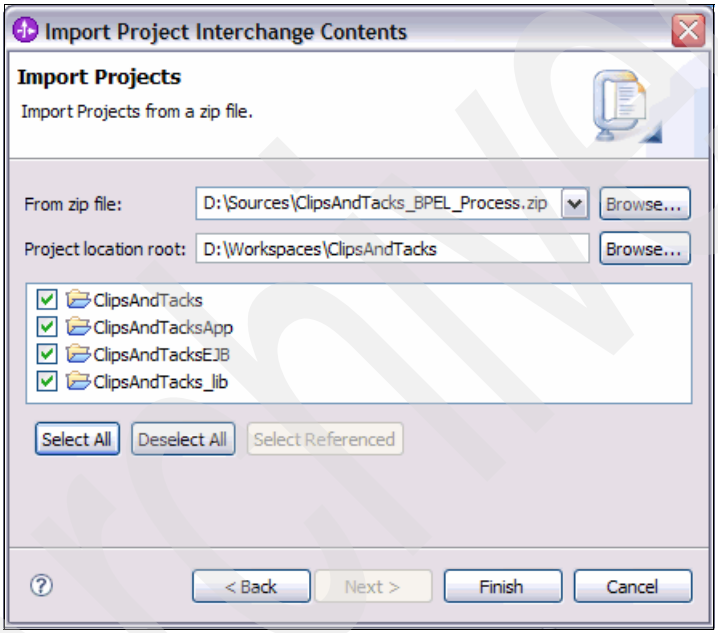


Figure 5-5 Selecting the ClipsAndTacks EAR file

Archive

5.3.3 Exploring the business process

In the previous section, we imported the ClipsAndTacks Order Handling process into a WebSphere Integration Developer workspace. As shown in Figure 5-6, the Business Integration explorer shows two folders:

- ClipsAndTacks

  This folder contains the implemented business process, one business rule, and implementations for external services.

- ClipsAndTacks_lib

  This folder contains all required data type and interface definitions.

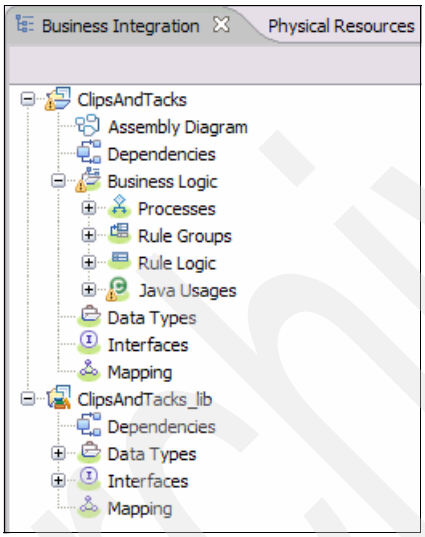


Figure 5-6 Business Integration explorer

Archived

Assembly diagram

Figure 5-7 shows the assembly diagram of the Order Handling process. It shows the business process (OrderHandling) on the left side and all services that are invoked by the process on the right side. It shows three Java components (VerifyCustomerCredit, CancelOrder, UpdateDatabase) and one business rule (AutomaticApproval). Because the process starts by using the integrated Business Process Choreographer Explorer, an export component is not required. To simplify the scenario, the Java components consist of basic implementations.

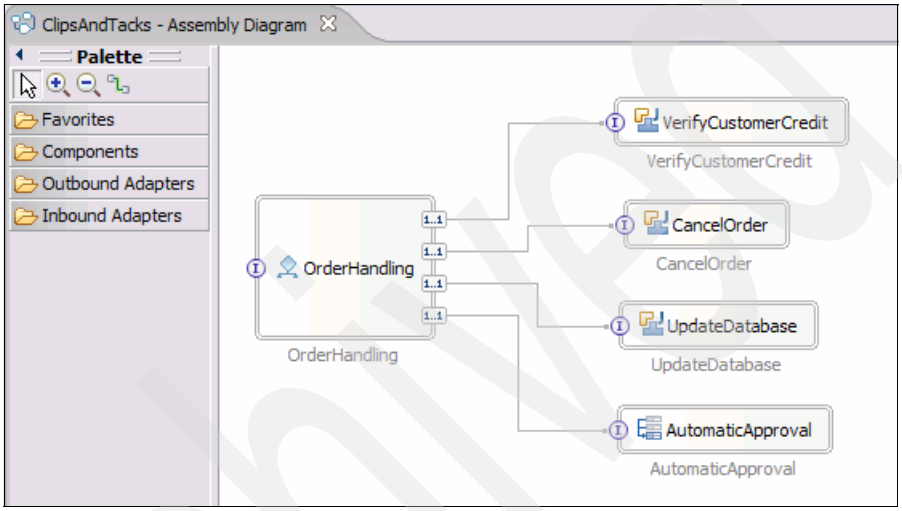


Figure 5-7 ClipsAndTacks Assembly Diagram

Implementations of external services

You can find the implementation of the Java components under **ClipsAndTacks** → **Business Logic** → **Java Usages** in the Business Integration Explorer. As mentioned previously, the Java components contain simplified implementations, which are explained in the following sections.

VerifyCustomerCredit

The VerifyCustomerCredit class contains the *verifyCustomer* method, which checks whether the total price of an order is higher than the available credit of the customer. In case it is, it sets the order status to REVIEW or to APPROVED.

CancelOrder

The CancelOrder class contains a *cancelOrder* method that creates a notification for the customer.

Archive

UpdateDatabase

The UpdateDatabase class contains the *updateDatabase* method, which returns a given order.

Business rule

The ClipsAndTacks Order Handling process uses a business rule to determine if an incoming order must be reviewed. The rule logic is under **ClipsAndTacks** → **Business Logic** → **Rule Logic**. As shown in Figure 5-8 on page 160, orders of gold customers are reviewed if the total price is higher than $1750. Silver customer orders are reviewed if the total price is higher than $1250. All other customer orders must be reviewed for a total price higher than $750. The business rule sets the order status of an order to REVIEW or AUTO-APPROVED.

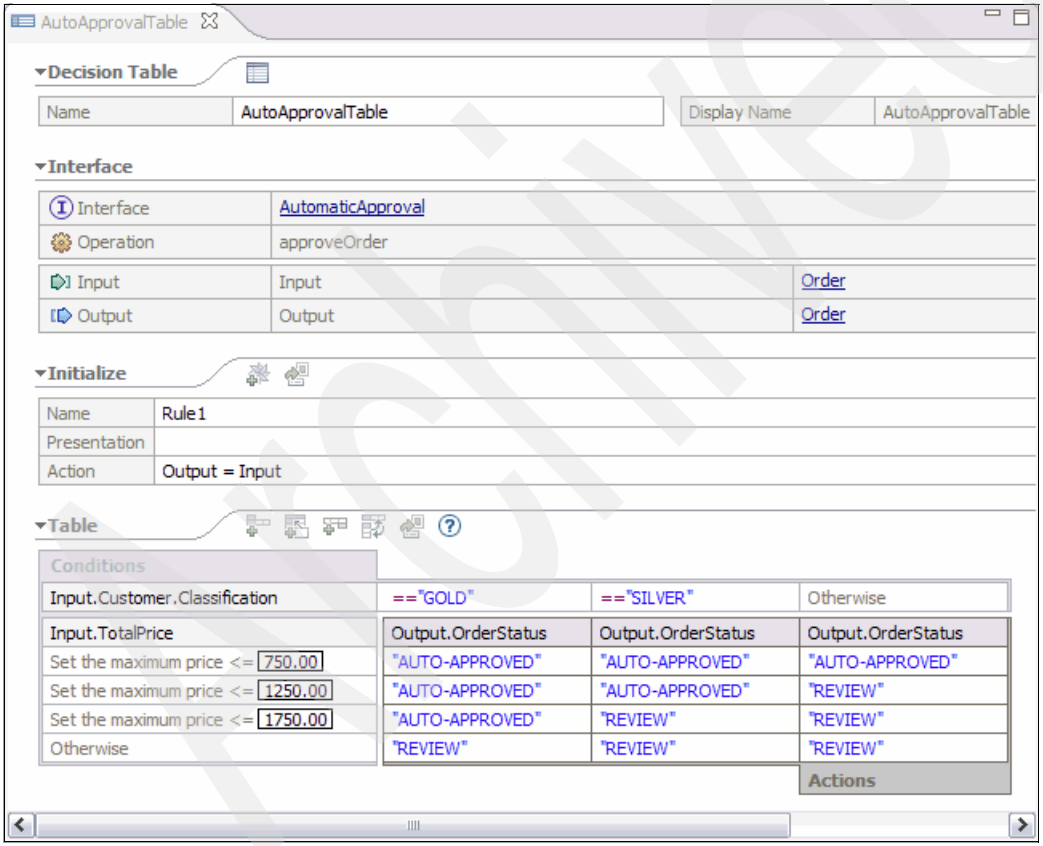


Figure 5-8 AutomaticApproval business rule

BPEL process

Figure 5-9 shows the BPEL process implementation of the ClipsAndTacks Order Handling process. The process is in the Business Integration view under **ClipsAndTacks** → **Business Logic** → **Processes**.

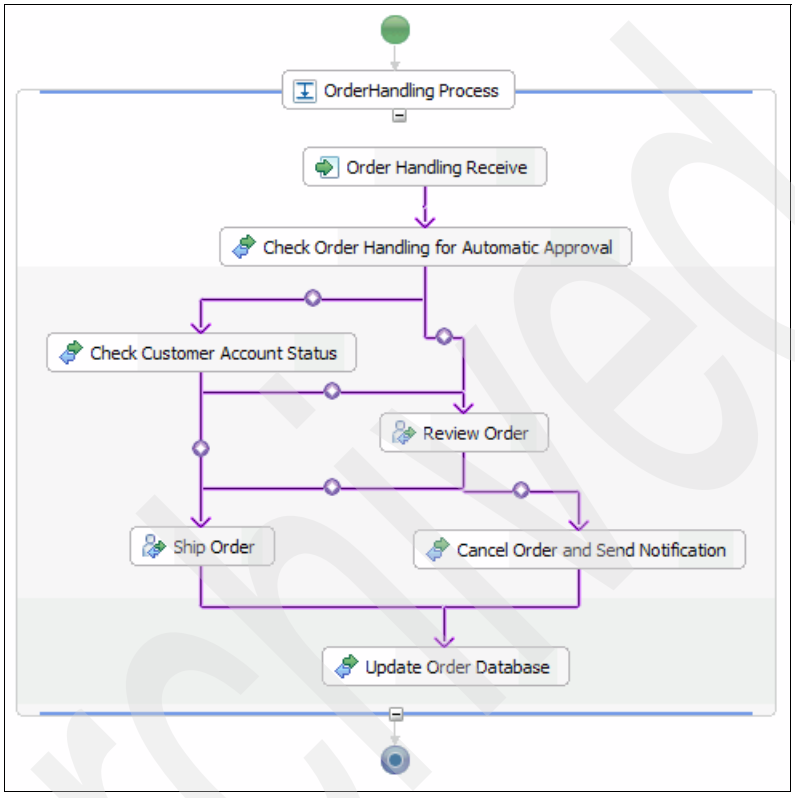


Figure 5-9 Order Handling BPEL process

The process consists of several activities, which are explained in the following sections.

Order Handling Receive

The Order Handling Receive activity is the start activity. It receives an order object as input and stores it in the OrderVariable variable of the process.

Check Order Handling for Automatic Approval

The Check Order Handling for Automatic Approval invoke activity invokes the business rule described in “Business rule” on page 160. In case the order status is set to REVIEW by the business rule, the process continues with the Review Order activity. Otherwise the Check Customer Account Status activity is called.

Archived

Check Customer Account Status

The Check Customer Account Status activity calls an external service that checks the customer account status. It sets the order status to APPROVED and navigates to the Ship Order activity in case the available credit of a customer is higher than the total price of the order. Otherwise the order status is set to REVIEW, and the process continues with the Review Order activity.

Review Order

The Review Order activity is a human task. The person who claims that task must decide whether an order is shipped or must be cancelled. The return value of this task is either APPROVED or DECLINED. In the first case, the process continues with the Ship Order activity. In the second case, the Cancel Order and Send Notification activity is called.

Ship Order

The Ship Order activity is a human task. The claimer of this task must provide input for a notification before finishing the order shipping.

Cancel Order and Send Notification

The Cancel Order and Send Notification invoke activity calls an external service that generates and returns a notification for a customer.

Update Order Database

The Update Order Database activity calls an external service.

5.3.4 Defining CEI events for the BPEL process

You must define which CEI events should be emitted at run time. For the monitoring requirements given by ClipsAndTacks, only a subset of business process events are of interest. The events are defined for the process and for some activities in the process.

Archive

The Properties view for the process and the process activity level contains an Event Monitor page (Figure 5-10) to define the events that are emitted to CEI at run time.

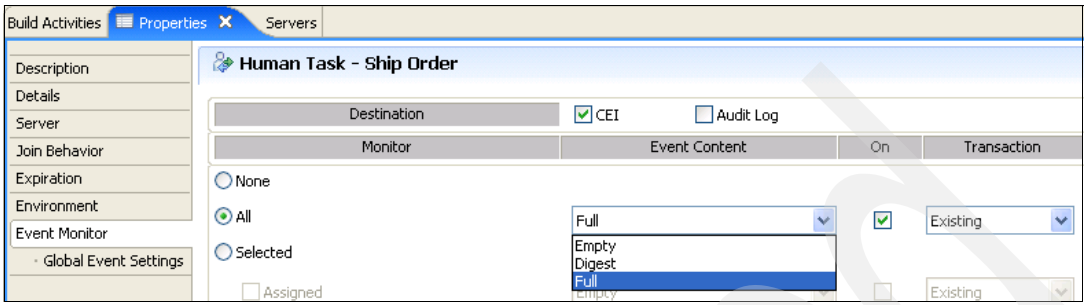


Figure 5-10 Event Monitor page

You can choose between the following Monitor options:

- None

  This option states that no events are emitted to CEI.

- All

  Selecting All defines that all possible events are emitted at run time. Due to performance reasons, we do not recommend enabling that option in a production environment. Whenever we select it in this scenario, we done so for simplification and demonstration reasons.

- Selected

  This option enables the selection of a specific set of events that are emitted to the CEI.

In the Event Content column, the content of an emitted event can be defined. The following options are not all available to all event types:

- Empty

  No payload is emitted with the event.

- Digest

  The payload description is emitted only.

- Full

  The full payload is emitted with the event.

Archived

In the Transaction column, you can furthermore define the transaction setting of an event:

- Existing

  If this option is selected, the event is sent in the current transaction. If the transaction fails, the event is not generated.

- New

  If this setting is chosen, the event is sent in a new transaction.

- Default

  If this setting is chosen, a default transaction value supported by the underlying event infrastructure (Existing or New) is used.

On the Global Event Settings page (Figure 5-11), you can choose the format of the emitted events. In V6.0.2, event definitions are stored in a .cbe file by using a nonstandard file format. In V6.1, an XML Schema Definition (XSD) is used to define events, while the event payload is emitted as XML content within the Common Base Event. In this scenario, the default setting WebSphere Business Monitor 6.1 format is used.

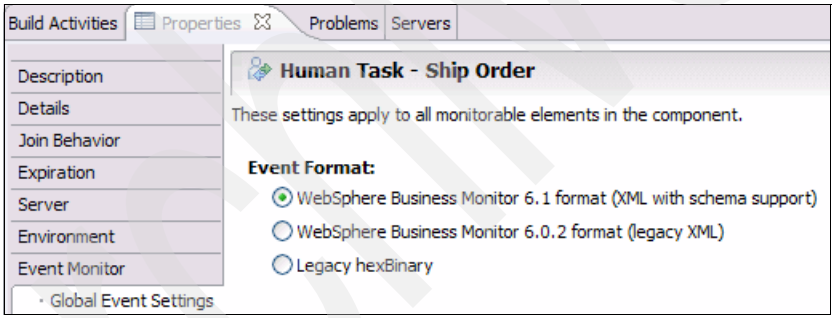


Figure 5-11 Global Event Settings page

Archive

Order Handling process

The monitoring context is defined on the process level. Therefore, events must be emitted when a process is started to instantiate a monitoring context. The monitoring context instance is terminated when a process is finished.

To turn on event monitoring for the Order Handling process:

1. Double-click the **OrderHandling** process to open the process in the BPEL editor (Figure 5-12).

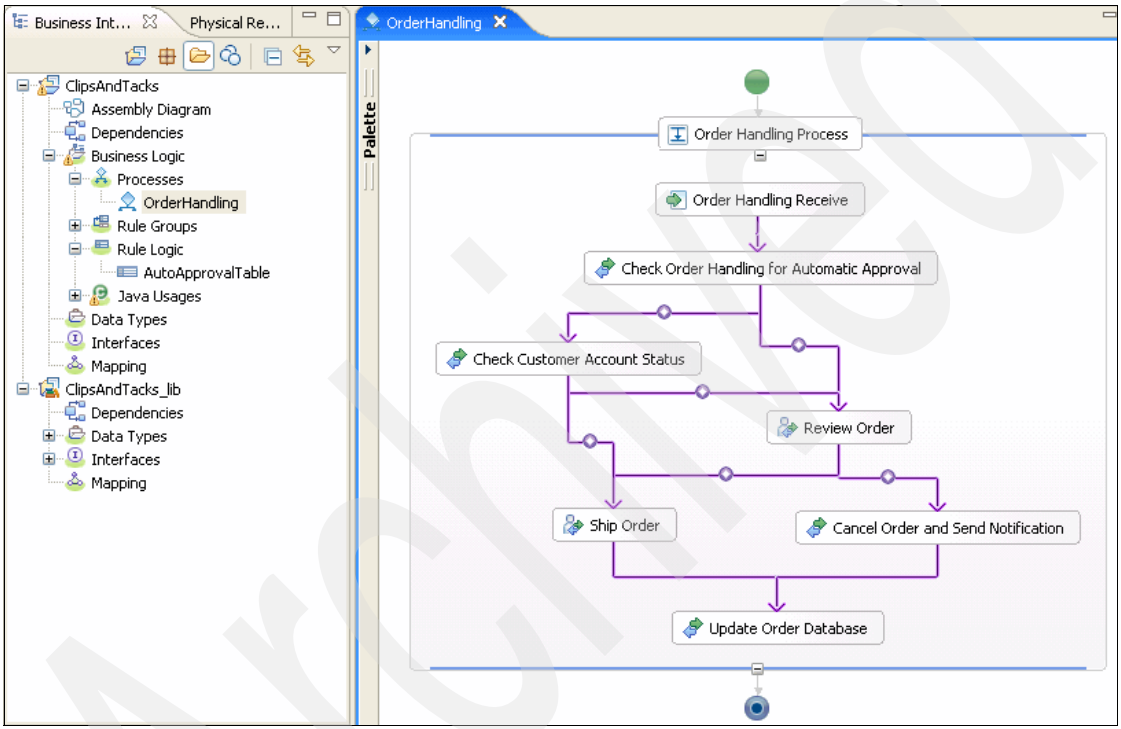


Figure 5-12 Order Handling process view

2. Click the white background of the process (outside of the flow element).

Archived

3. Click the **Properties** tab of the Order Handling process and then click the **Event Monitor** subtab. Select **All** (Figure 5-13).

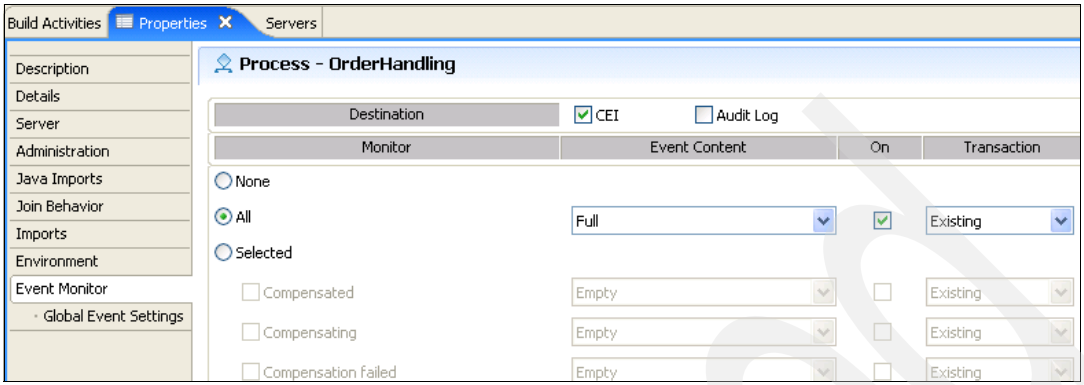


Figure 5-13 Process event selection

4. Press Ctrl+S to save your work.

After selecting events on the process level, a small yellow flag icon (Figure 5-14) is shown in the BPEL editor. The flag indicates that events are defined for the process.

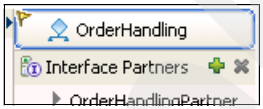


Figure 5-14 Event icon

Order Handling Receive activity

The monitoring context will contain metrics that capture values of the OrderVariable variable that is input into the process. The first time the content of this variable is accessible is at the exit of the Order Handling Receive activity. Therefore, an event must be emitted at this point. To access the variable content, Event Content must be set to *full* as explained in the following steps:

Archived

1. Select the **Order Handling Receive** activity in the process.
2. Click the **Properties** tab and then click the **Event Monitor** subtab (Figure 5-15). Click **Selected** and select the **Exit** check box.

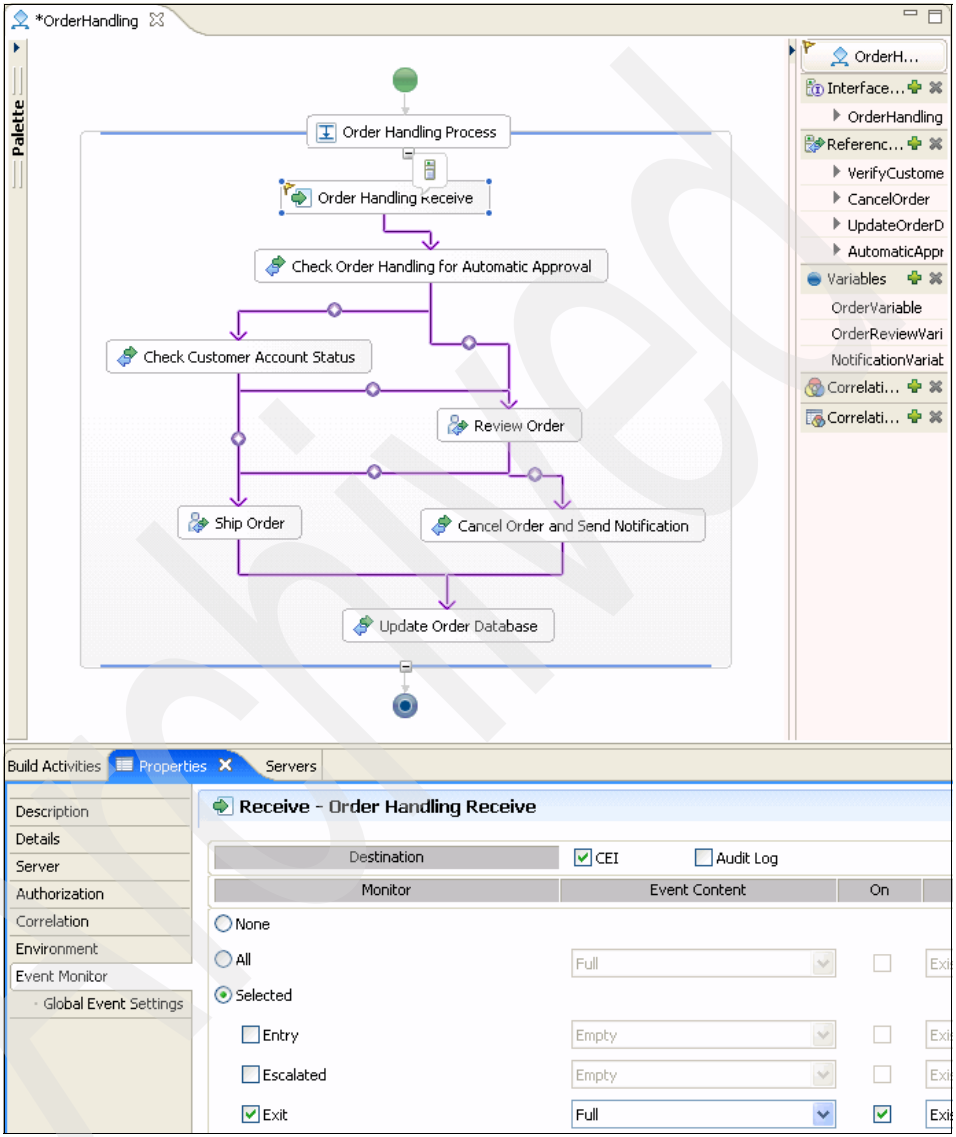


Figure 5-15 Order Handling Receive activity event selection

3. Press Ctrl+S to save your work.

Ship Order activity

ClipsAndTacks is interested in average order fulfillment duration and percentage of shipped orders. To calculate these KPIs, they must know when an order is shipped, and therefore, an event must be emitted when the Ship Order activity is exited. To define this setting:

1. Select the **Ship Order** activity in the process.
2. Click the **Properties** tab and then click the **Event Monitor** subtab (Figure 5-16). Click **Selected** and select the **Exit** check box.

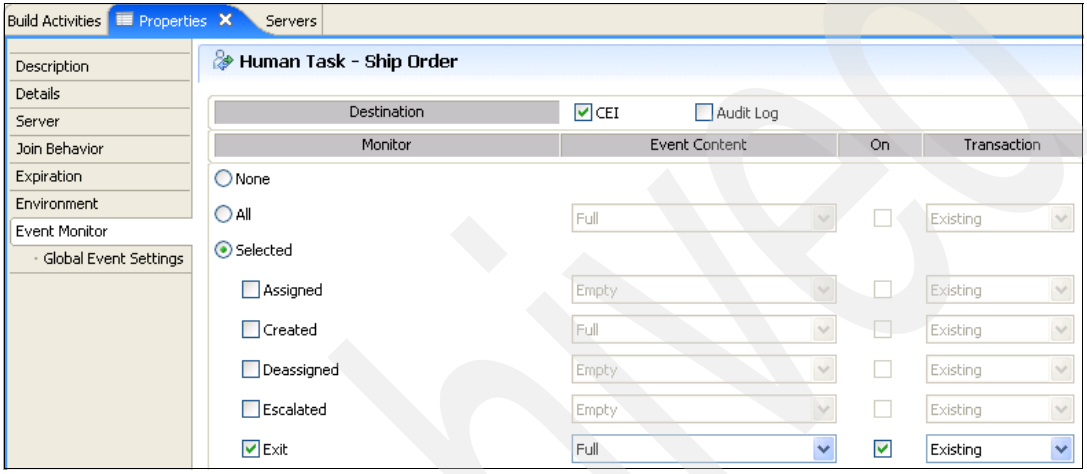


Figure 5-16 Ship Order activity event selection

3. Press Ctrl+S to save your work.

Archived

Cancel Order and Sent Notification activity

To determine when an order is cancelled, the exit event of this activity must be triggered. To trigger this event:

1. Select the **Cancel Order and Sent Notification** activity in the process.
2. Click the **Properties** tab and then click the **Event Monitor** subtab (Figure 5-17). Click **Selected** and select the **Exit** check box.

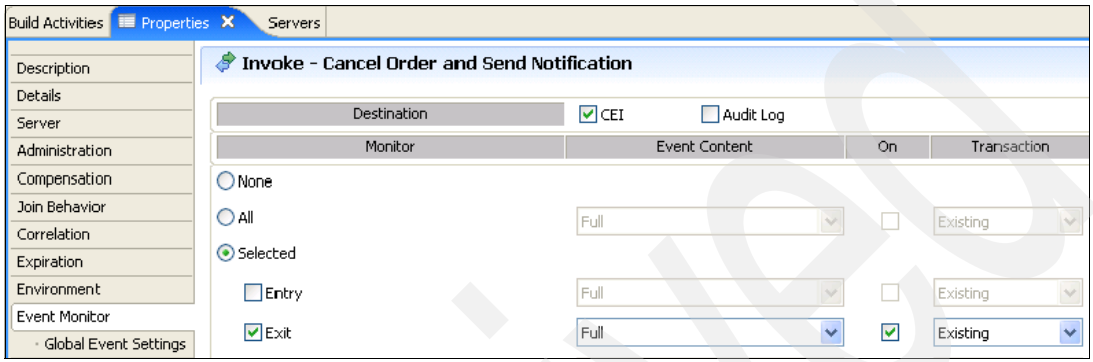


Figure 5-17 Cancel Order and Send Notification activity event selection

3. Press Ctrl+S to save your work.

5.3.5 Developing the monitor model by using the WebSphere Business Monitor Toolkit

The ClipsAndTacks monitor model is auto-generated by a wizard based on the ClipsAndTacks Order Handling process. The generated monitor model contains the monitoring context instance creation, termination, and correlation information based on the events that are defined in the process. Further monitor model components and the KPI model must be implemented.

Archive

Generating the monitor model

The auto-generation of the monitor model is based on the ClipsAndTacks Order Handling process. To create the model:

1. In the Business Integration view, right-click the **ClipsAndTacks** folder and select **Monitor Tools** → **Generate Monitor Model**.
2. In the Generate Monitor Model window (Figure 5-18), for Target monitor project, type `ClipsAndTacksMonitoring` and click **New Project**.

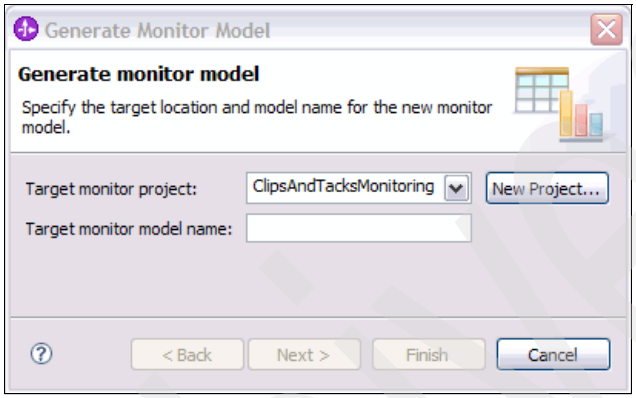


Figure 5-18 Selecting a target monitor project

3. In the New Business Monitoring Project window (Figure 5-19), click **Finish**.

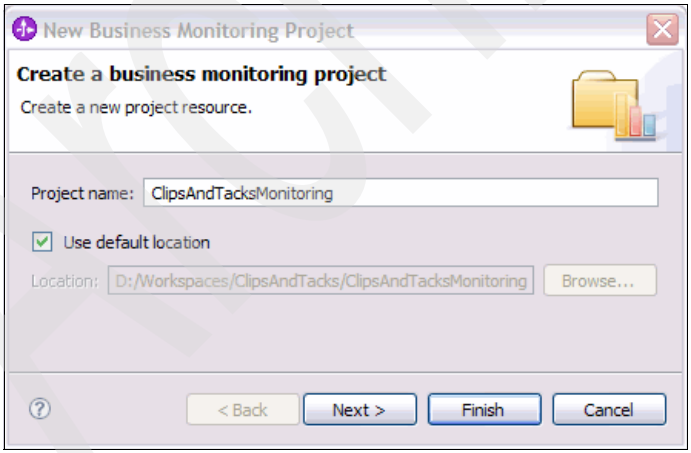


Figure 5-19 Creating a monitoring project

4. In the Generate Monitor Model window (Figure 5-20), for Target monitor model name, type `ClipsAndTacksMM` and click **Next**.

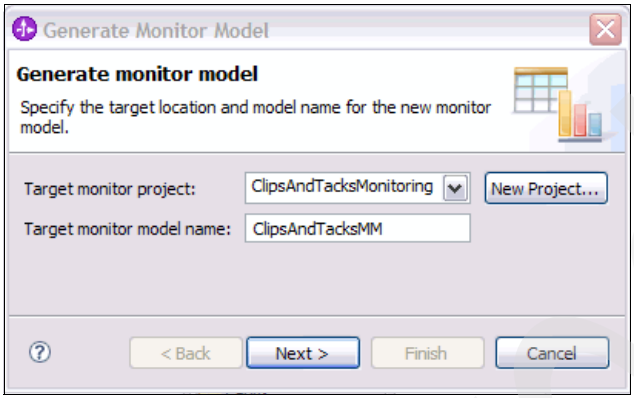


Figure 5-20 Selecting a target monitor model name

Archived

5. In the Generate Monitor Model – Choose what to monitor window (Figure 5-21), select the events to be reflected in the monitor model for the process:

   a. In the Event Source pane, expand **ClipsAndTacks** → **ClipsAndTacksModule** → **OrderHandling** and click **OrderHandling**.

   b. The MonitoringTemplates page shows predefined metrics and KPI templates. When they are enabled automatically, all required events, triggers, and metrics are selected. Select the **Start Time** and **End Time** template.

   c. Click the **Emitted Events** tab to check which events will be added to the monitor model.

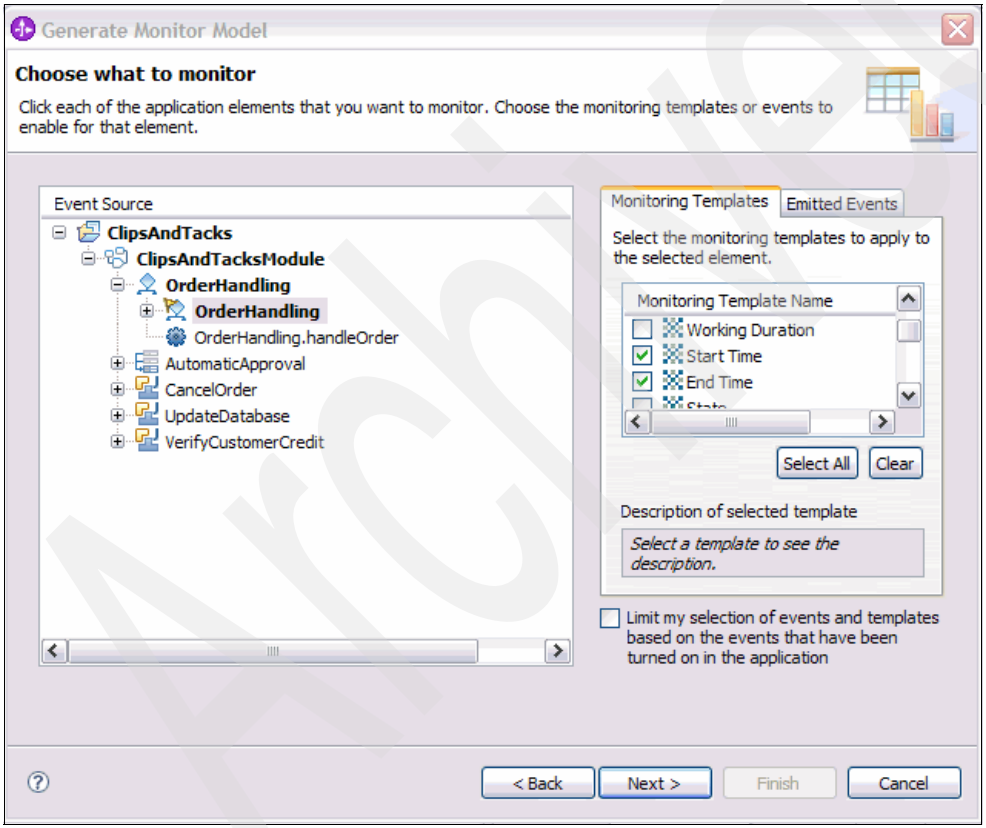


Figure 5-21 Selecting the process monitoring templates

d. For the process activities, no templates are selected on the Monitoring Templates page, but the events defined in WebSphere Process Server are selected directly on the Emitted Events page. Select the monitor model events for the process activities:

   i. In the Event Source pane, expand **ClipsAndTacks** → **ClipsAndTacksModule** → **OrderHandling** → **OrderHandling** → **Order Handling Process** and click **Order Handling Receive**.

   ii. Click the **Emitted Events** tab (Figure 5-22).

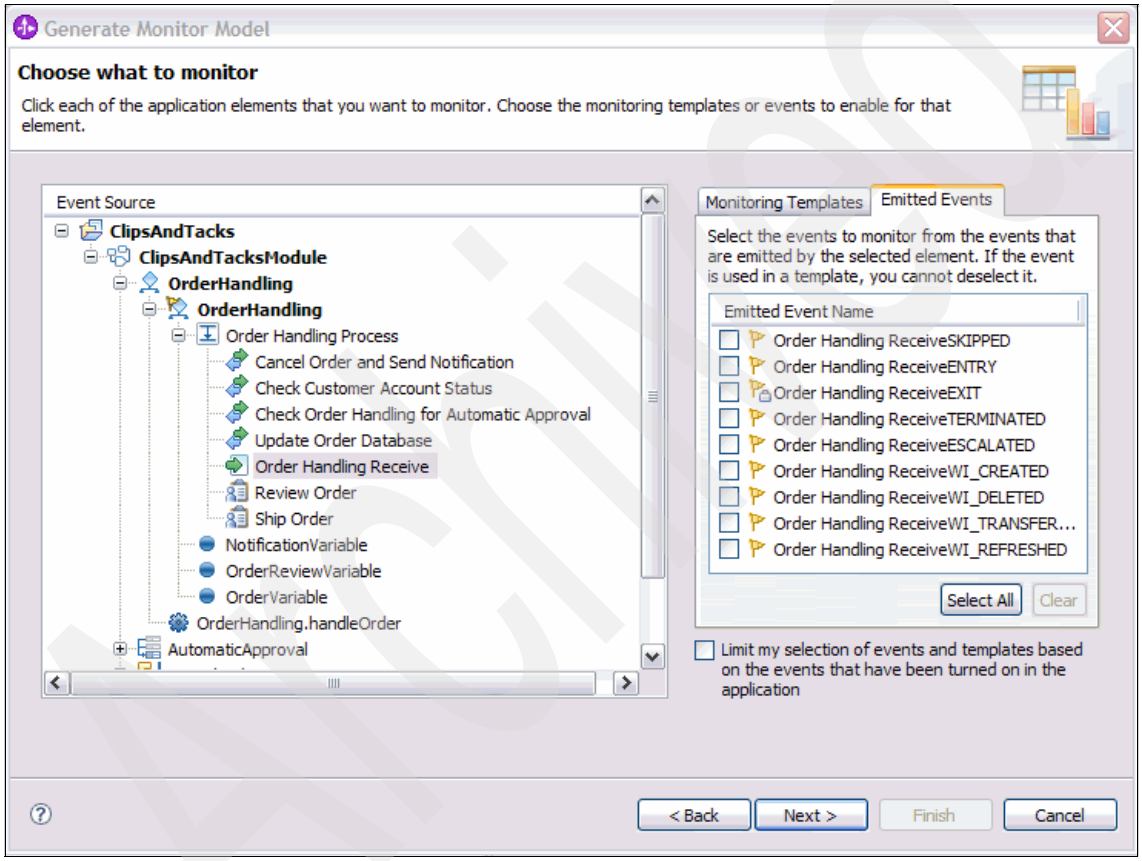


Figure 5-22 Selecting the Order Handling Receive activity emitted events

iii. Select the **Order Handling ReceiveEXIT** event.

By default, the Emitted Events page shows all possible events. Select the **Limit my selection of events and templates based on the events that have been turned on in the application** to minimize the possible event choices (Figure 5-23).

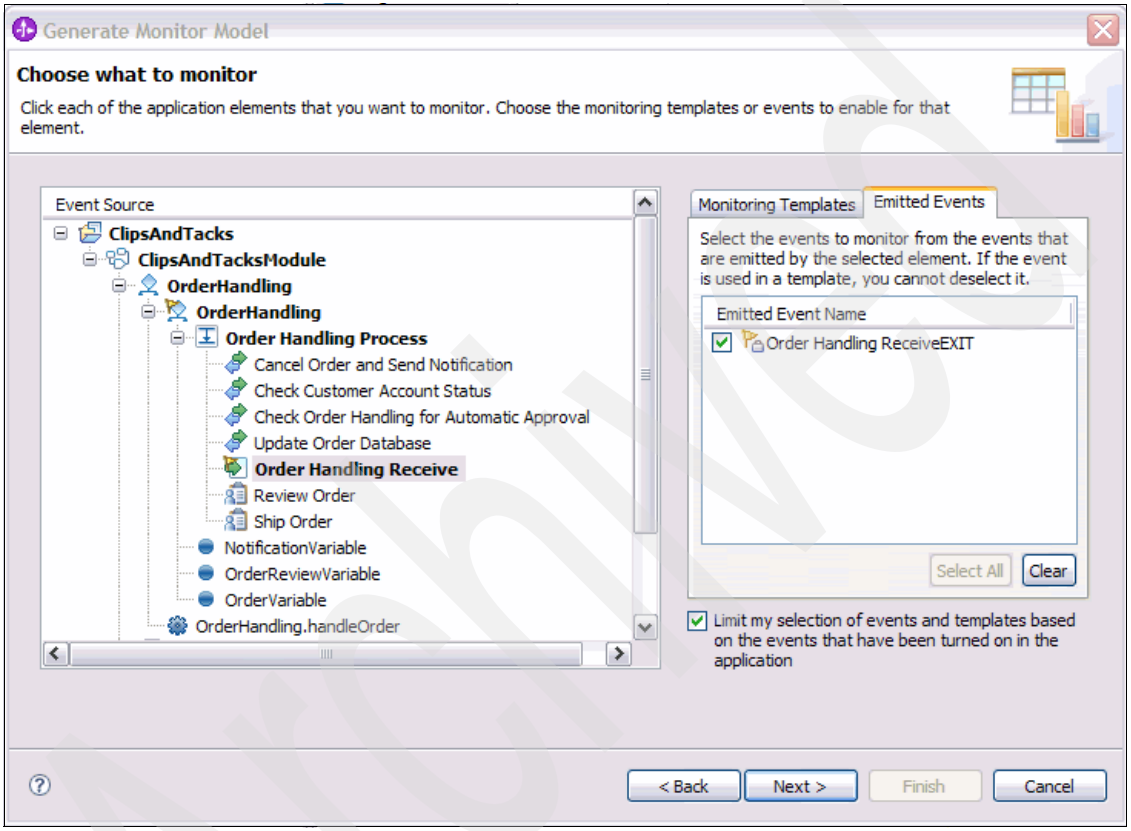


Figure 5-23 Limited Emitted Events list

e. Repeat step d on page 173 for the **Cancel Order and Send Notification** and **Ship Order** activities.

f. Click **Next**.

6. In the Generate Monitor Model – Choose how to monitor window (Figure 5-24), define the implementation of the monitor model. The monitor model is simplified by flattening the structure, so that only one monitoring context exists for the process. Create event groups to organize the events. Provide structure in the monitor model by changing the implementation for the activities to **Event group**. Click **Next**.

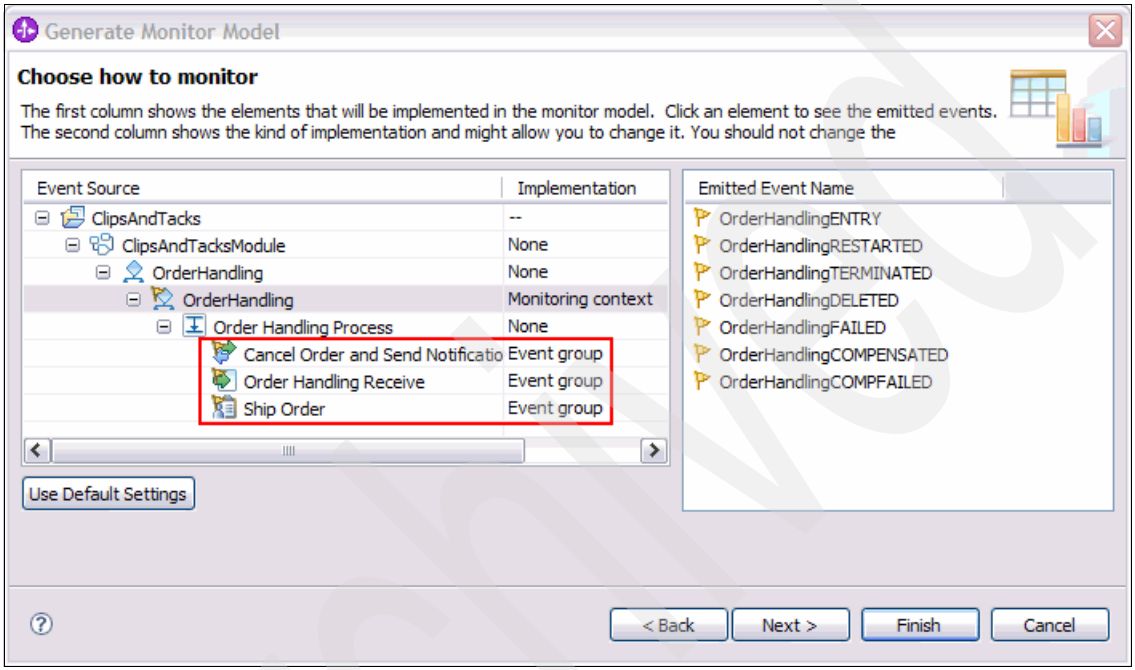


Figure 5-24 Monitor Model implementation

Archived

7. In the Generate Monitor Model – Preview the monitor model window (Figure 5-25), review the monitor model. Click **Back** to make changes as necessary or click **Finish** to generate the monitor model.

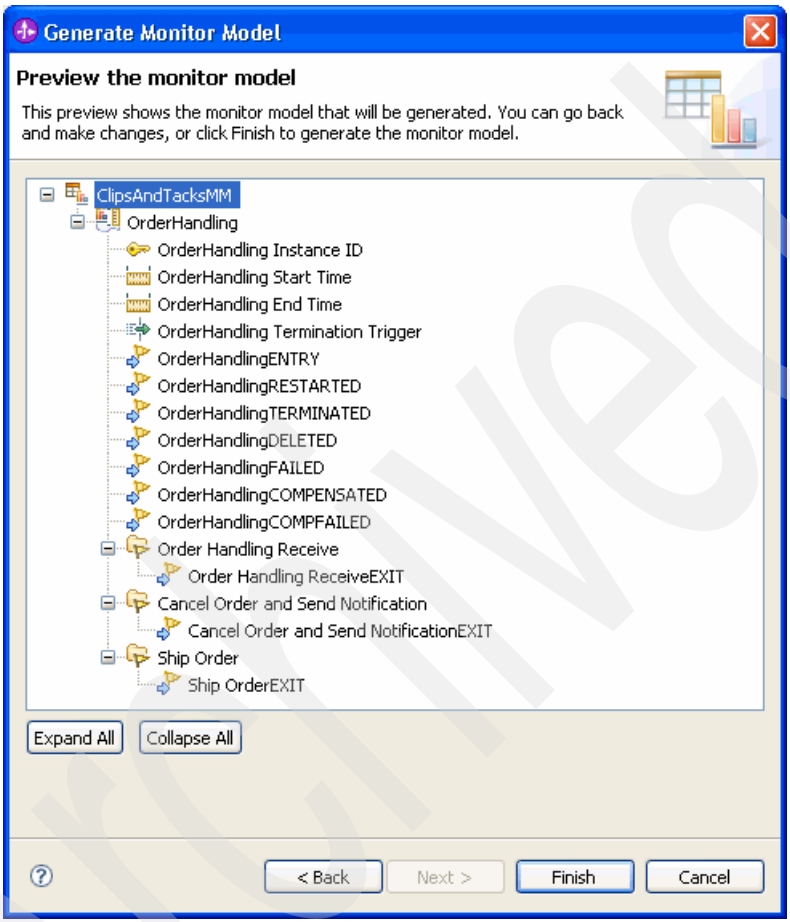


Figure 5-25 Monitor Model preview

8. When prompted, click **Yes** to switch to the Business Monitoring perspective.
9. When prompted, do not launch *Get Started*.

Archived

Exploring the generated monitor model

After the monitor model generation, the created file opens automatically in the Monitoring perspective (Figure 5-26).

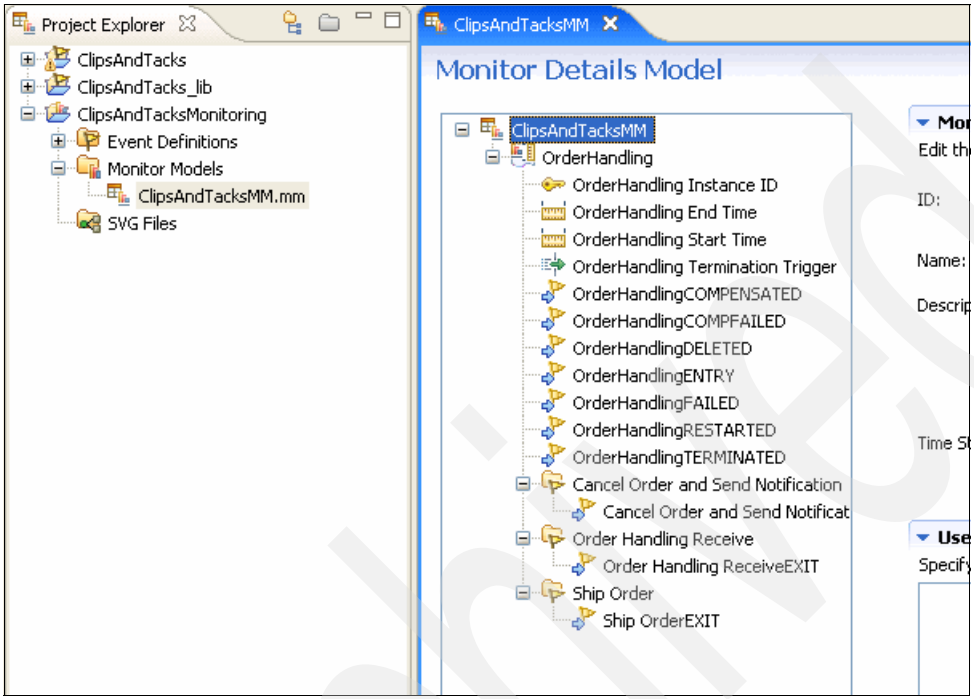


Figure 5-26 Generated Monitor Model

The monitor model contains one monitoring context for the process (OrderHandling). The key for each monitor context instance is the process instance ID (OrderHandling instance ID). Because the process Start Time and End Time template were selected, WebSphere Integration Developer automatically generates metrics for both times and a trigger (OrderHandling Termination Trigger) to terminate a monitoring context instance (Figure 5-27).

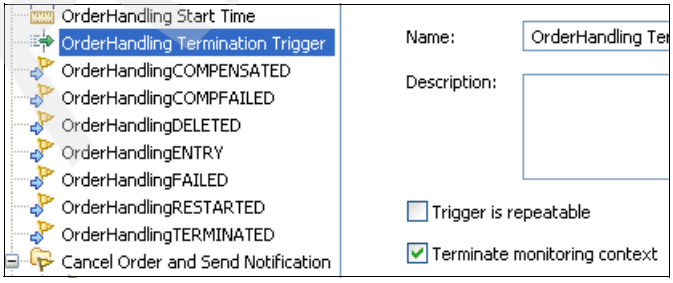


Figure 5-27 Terminating a monitoring context

The two metrics and the trigger are based on all possible process start events (OrderHandlingENTRY, OrderHandlingRESTARTED) and process termination events (OrderHandlingCOMPENSATED, OrderHandlingCOMPFAILED, OrderHandlingDELETED, OrderHandlingFAILED, OrderHandlingTERMINATED). Furthermore three event groups generated for the exit events of the process activities are shown (Figure 5-28).

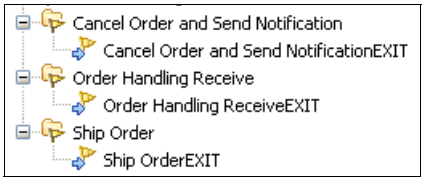


Figure 5-28 Generated event groups

To calculate KPIs, the monitor model must be extended by the following components:

- Trigger
- Stopwatch
- Metric
- Counter
- KPI
- Dimension
- Alert
- Visual model

Creating triggers and a stopwatch

To fill metrics and stopwatches with values, you must create several triggers based on the generated events as explained in the following section. You must also create a stopwatch, which ClipsAndTacks uses to measure the order fulfillment duration.

Archived

New Order Trigger

This trigger invoked whenever a new Order Handling process is started. To create the New Order Trigger:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Trigger** (Figure 5-29).

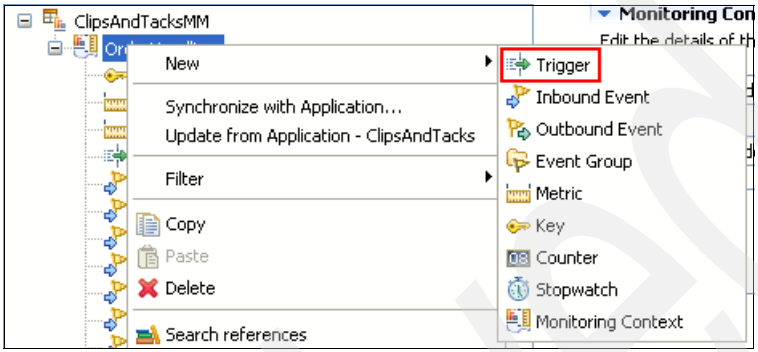


Figure 5-29 Creating a new trigger

2. In the Create New Trigger window (Figure 5-30), for Name, type `New Order Trigger` and click **OK**.

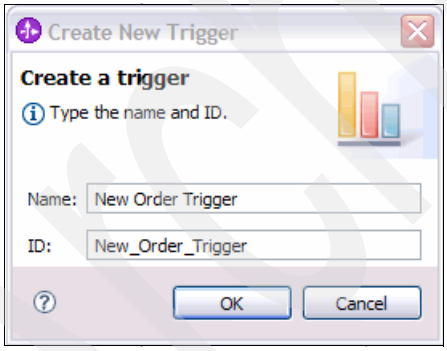


Figure 5-30 Creating a New Order trigger

Archived

3. In the Monitor Details Model panel (Figure 5-32 on page 181), under Trigger Sources:
   a. Click the **Add** button.
   b. In the Select Trigger Source window (Figure 5-31), select **Other source type** and select **OrderHandlingENTRY**. Then click **OK**.

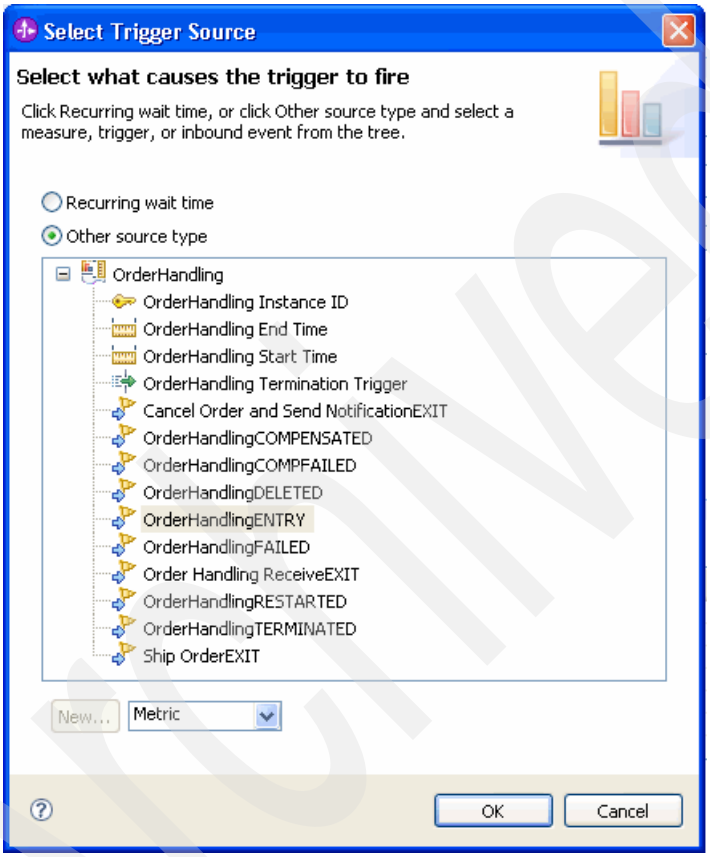


Figure 5-31 Selecting the New Order Trigger event

Archived

c. Click the **Add** button again.

d. In the Select Trigger Source window, click **Other source type** and select **OrderHandlingRESTARTED**. Then click **OK**. Figure 5-32 shows the completed Trigger Sources section.

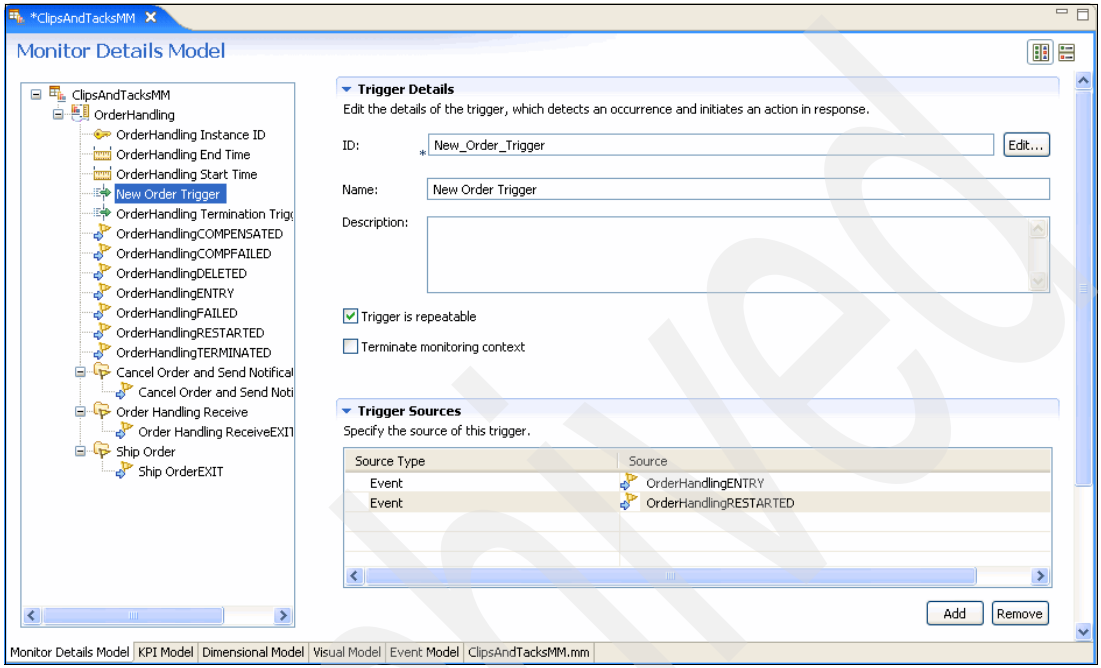


Figure 5-32 New Order Trigger overview

4. Press Ctrl+S to save your work.

Order Shipped Trigger

The Order Shipped Trigger indicates when an order has been shipped. To create the Order Shipped Trigger:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Trigger**.
2. In the Create New Trigger window, for Name, type `Order Shipped Trigger` and click **OK**.

Archived

3. In the Monitor Details Model panel (Figure 5-33), under Trigger Sources:
   a. Click the **Add** button.
   b. In the Select Trigger Source window, click **Other source type** and select **ShipOrderEXIT**. Then click **OK**.

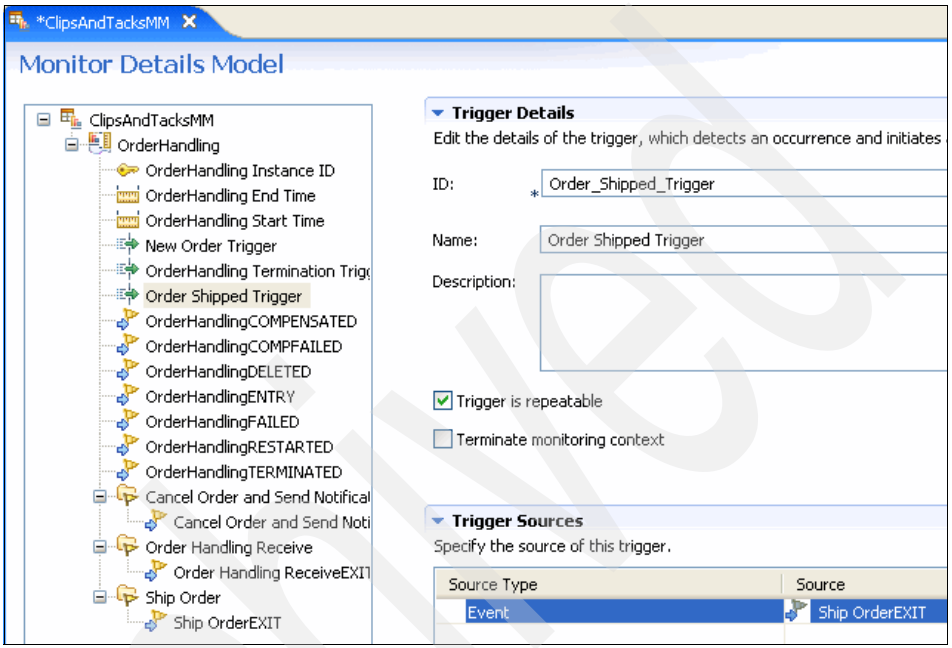


Figure 5-33 Order Shipped Trigger overview

4. Press Ctrl+S to save your work.

Order Cancelled Trigger

The Order Cancelled Trigger indicates when an order has been cancelled. To create the Order Cancelled Trigger:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Trigger**.
2. In the Create New Trigger window, for Name, type `Order Cancelled Trigger` and click **OK**.

Archived

3. In the Monitor Details Model panel (Figure 5-34), under Trigger Sources:
   a. Click the **Add** button.
   b. In the Select Trigger Source window, click **Other source type** and select **Cancel Order and Send NotificationEXIT**. Then click **OK**.

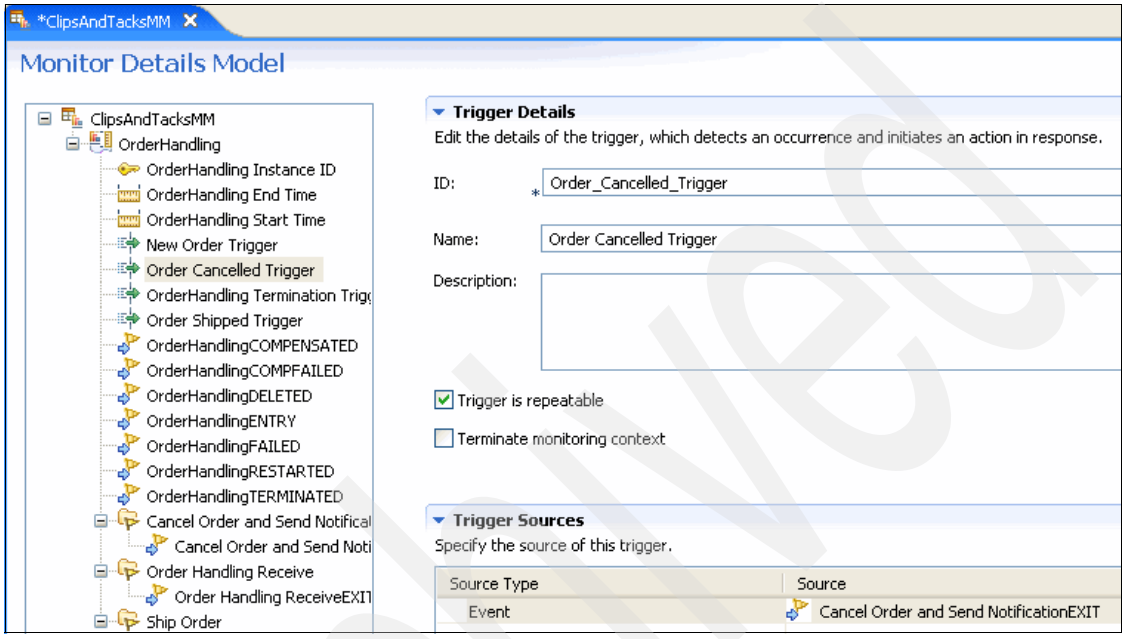


Figure 5-34 Order Cancelled Trigger overview

4. Press Ctrl+S to save your work.

Order Received Trigger

The Order Received Trigger indicates that the Order Handling Receive activity of the process is finished. It is used by a metric later. To create the Order Received Trigger:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Trigger**.
2. In the Create New Trigger window, for Name, type `Order Received Trigger` and click **OK**.

Archived

3. In the Monitor Details Model panel (Figure 5-35), under Trigger Sources:
   a. Click the **Add** button.
   b. In the Select Trigger Source window, click **Other source type** and select **Order Handling ReceiveEXIT**. Then click **OK**.

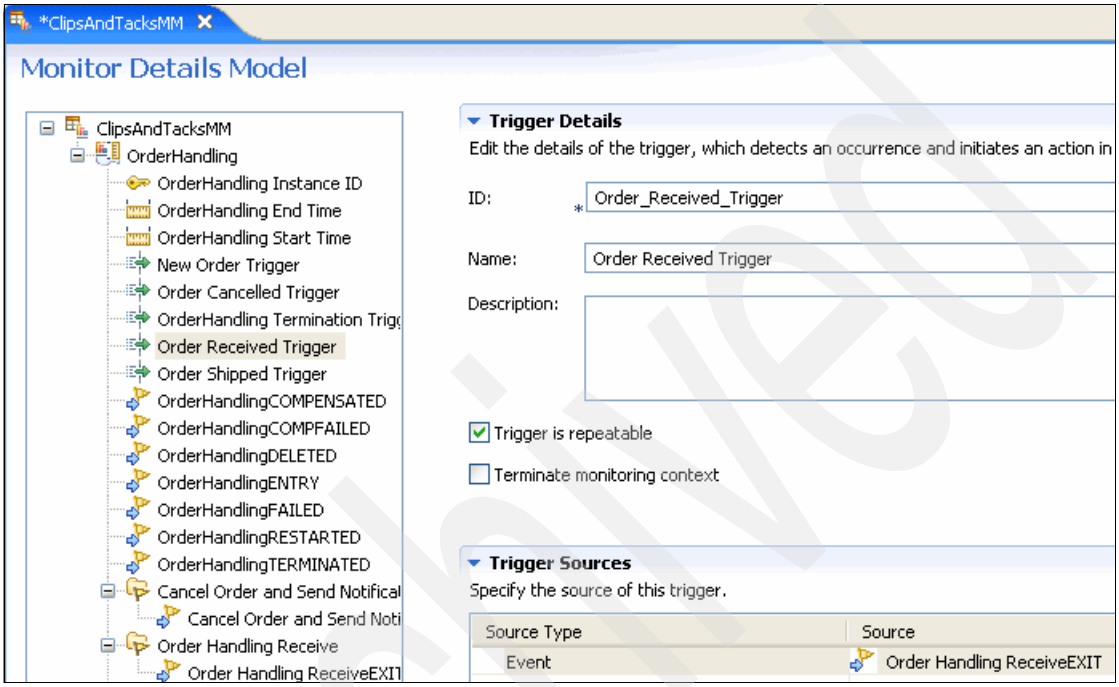


Figure 5-35 Order Received Trigger overview

4. Press Ctrl+S to save your work.

Creating a stopwatch

ClipsAndTacks want to check the average order fulfillment duration. Therefore they must know the time a process instance takes from process start to either order shipment or cancellation.To create a stopwatch to measure that time:

Archived

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Stopwatch**.
2. In the Create New Stopwatch window (Figure 5-36), for Name, type `Order Fulfillment Duration` and click **OK**.

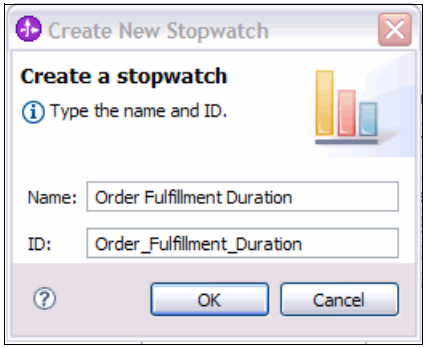


Figure 5-36 Creating a stopwatch

3. Under Stopwatch Controls:
   a. Click **Add**.
   b. Select **OrderHandling** → **New Order Trigger** and click **OK**.
   c. In the Resulting Action column, accept the default setting of **Start**.
   d. For Stopwatch Controls, click **Add**, select **OrderHandling** → **Order Cancelled Trigger** and click **OK**.
   e. In the Resulting Action column, change the default setting of *Start* to **Stop** for this trigger. Click the second row in the Resulting Action column (Figure 5-37) and select **Stop** from the list.

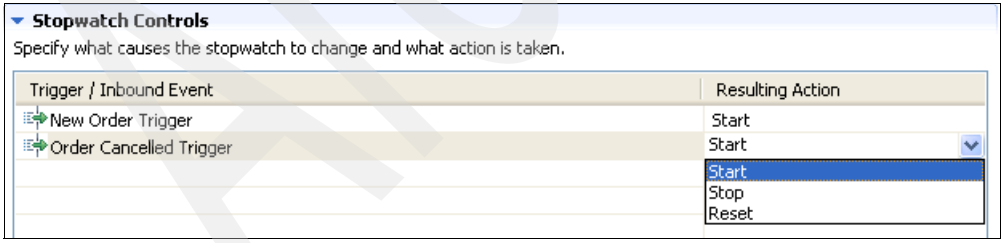


Figure 5-37 Stopwatch Resulting Action Selection

   f. Click **Add** again.
   g. Select **OrderHandling** → **Order Shipped Trigger** and click **OK**.

Archived

h. In the Resulting Action column, change the default setting of *Start* to **Stop** for this trigger. Click the second row in the Resulting Action column (Figure 5-38) and select **Stop** from the list.

4. Press Ctrl+S to save your work.

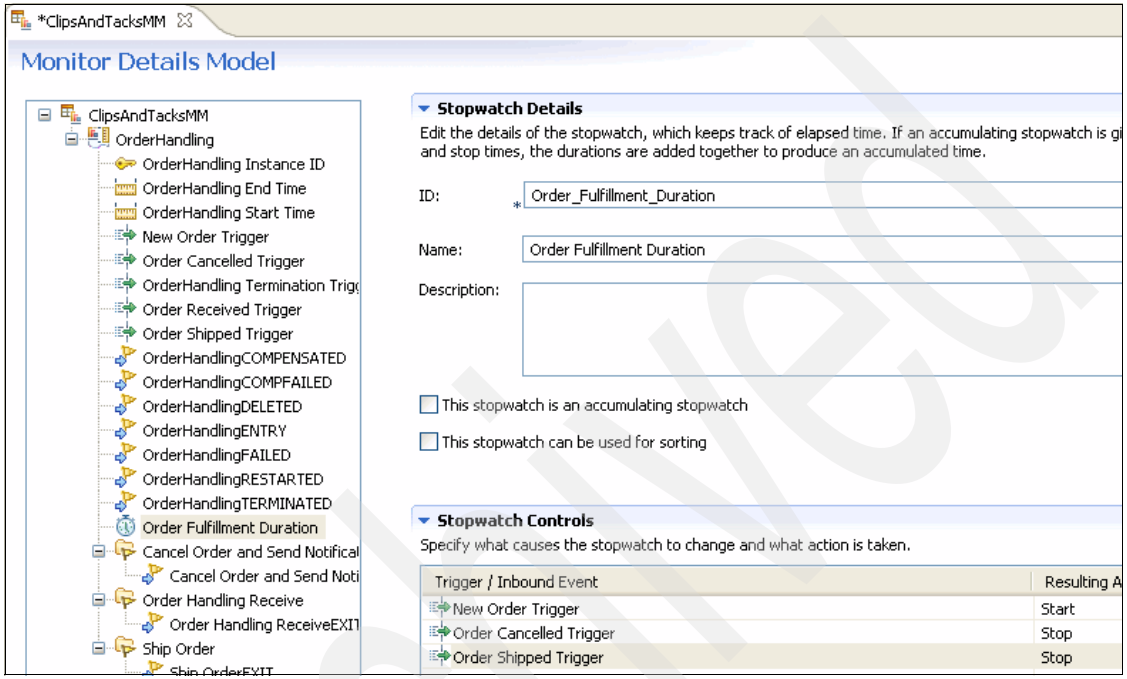


Figure 5-38 Order Fulfillment Duration Stopwatch overview

Late Order Fulfillment Event trigger

The Late Order Fulfillment Event trigger is invoked by a self-defined condition. It is required for an alert indicating that an order took longer than three days and is based on the stopwatch created in “Creating a stopwatch” on page 184. To create the Late Order Fulfillment Event trigger:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Trigger**.
2. In the Create New Trigger window, for Name, type `Late Order Fulfillment Event` and click **OK**.

Archived

3. Under Trigger Sources (Figure 5-39):
   a. Click the **Add** button.
   b. Select **Recurring wait time** and click **OK**.
   c. For Trigger Condition, type `Order_Fulfillment_Duration ge xs:dayTimeDuration('PT3M')`. Note that a trigger condition of *three minutes* is entered not *three days* as stated before. This is done for simplification reasons because it is hard to simulate an order that takes more than three days.

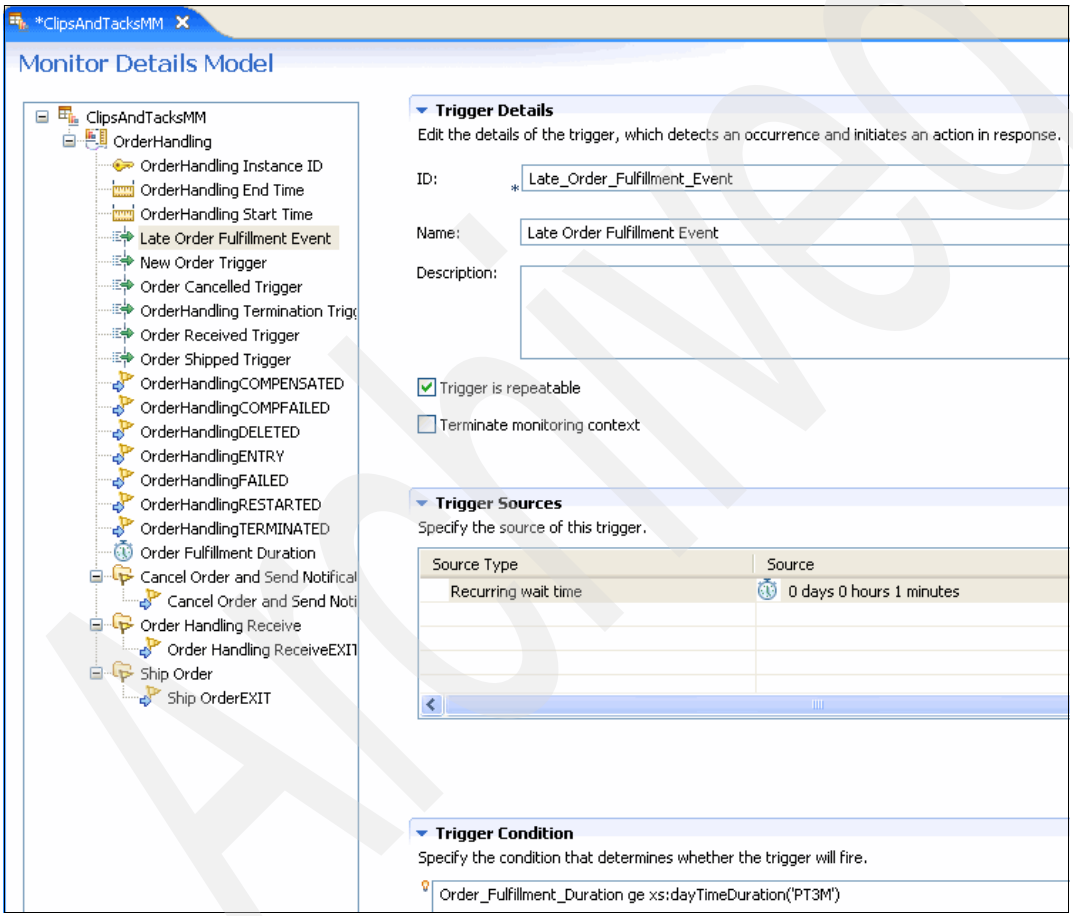


Figure 5-39 Late Order Fulfillment Event Trigger overview

4. Press Ctrl+S to save your work.

Archived

Creating metrics

Based on the trigger created in the previous section, we must develop all required metrics for the monitor model as explained in the sections that follow.

City

The city metric is required for a dimension that we create later. To create this metric:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Metric**.
2. In the next window, for Name, type `City`. For Type, select **String**. Click **OK**.
3. In the Monitor Details Model panel (Figure 5-41 on page 189):
   a. Under Metric Details:
      i. Select **A value is required for this metric**.

         The editor requires this selection whenever a metric is used as a source for a dimension in the dimensional model.

      ii. For Default Value, type `'Unknown'` with the single quotation marks.
   b. Under Metric Value Expressions:
      i. Click the **Add** button.
      ii. In the first row of the Metric Value Expressions table, select the cell under the Trigger column and click the button in the cell.
      iii. In the Select a Trigger window (Figure 5-40), select **Trigger type**. Then expand **OrderHandling** and select **Order Receive Trigger**. Click **OK**.

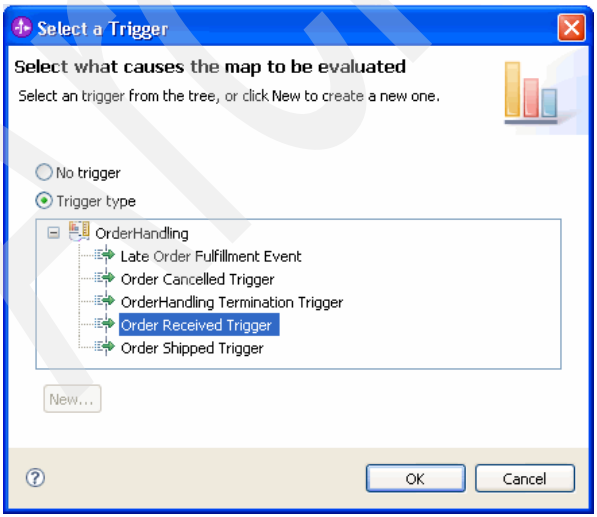


Figure 5-40 Trigger selection for the City metric

iv. In the first row of the Metric Value Expressions table, select the cell under the Expression column and click the button in the cell.

v. Type `Order_Handling_ReceiveEXIT/Input/Customer/City` and click **OK**.

Figure 5-41 shows the result of these settings In the Monitor Details Model panel.

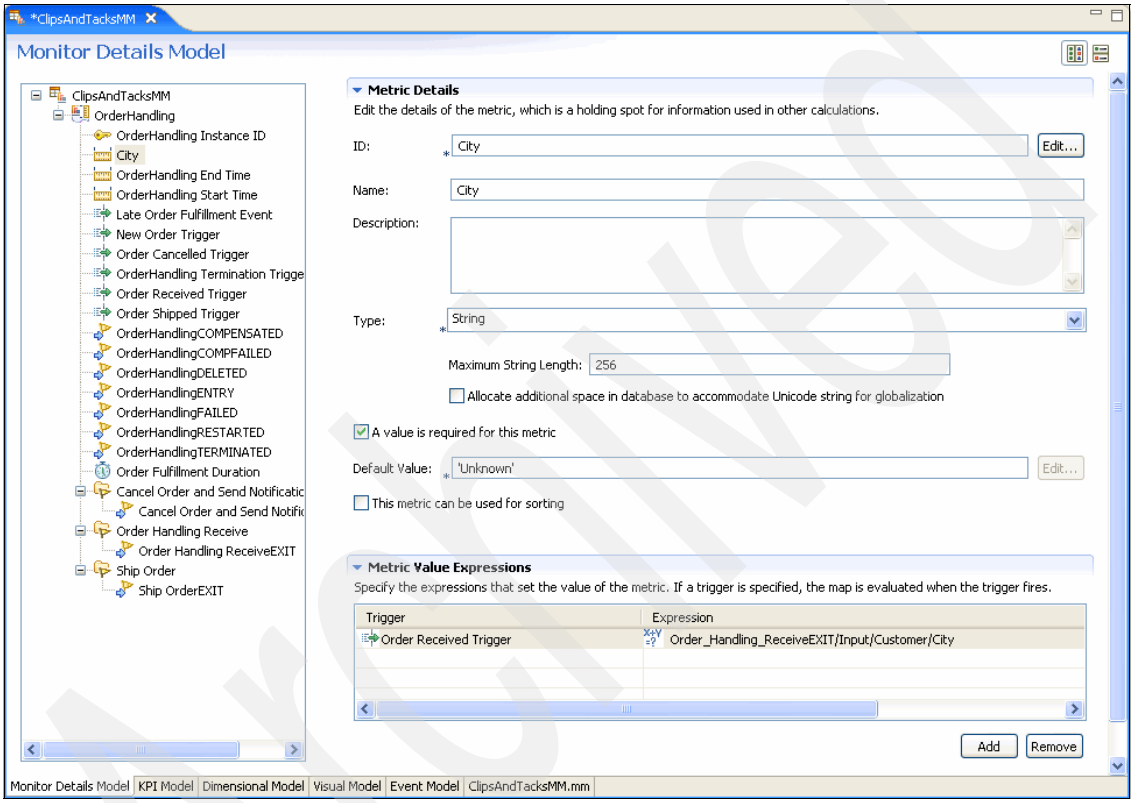


Figure 5-41 City metric overview

4. Press Ctrl+S to save your work.

Archived

Country

The country metric is also used in the dimension model. To create this metric:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Metric**.
2. In the next window, for Name, type `Country`. For Type, select **String**. Then click **OK**.
3. In the Monitor Details Model panel (Figure 5-42 on page 191):
   a. Under Metric Details:
      i. Select **A value is required for this metric**.
      ii. For Default Value, type `'Unknown'` with the single quotation marks.
   b. Under Metric Value Expressions:
      i. Click the **Add** button.
      ii. In the first row of the Metric Value Expressions table, select the cell under the Trigger column and click the button in the cell.
      iii. In the Select a Trigger window, select **Trigger type**. Then select **OrderHandling** → **Order Receive Trigger** and click **OK**.
      iv. In the first row of the Metric Value Expressions table, select the cell under the Expression column and click the button in the cell.
      v. Type `Order_Handling_ReceiveEXIT/Input/Customer/Country`, and click **OK**.

Archived

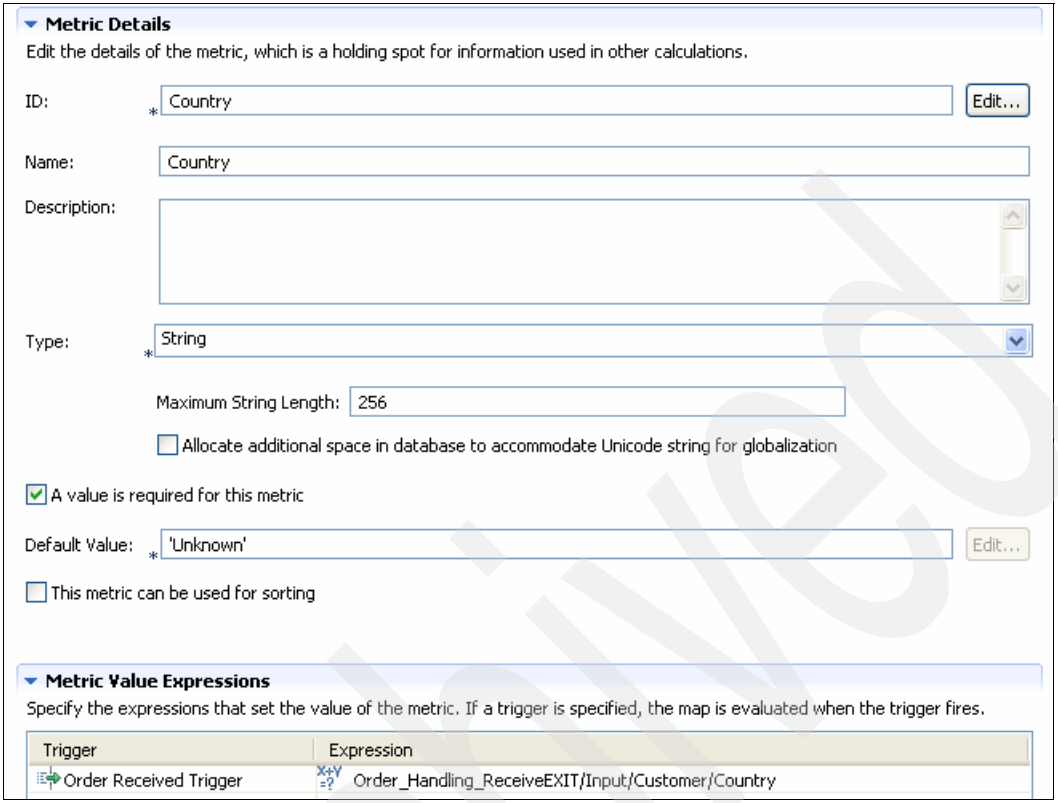


Figure 5-42 Country metric overview

4. Press Ctrl+S to save your work.

Order Status

The Order Status metric is used by the visual model that is created later. It indicates if an order instance is shipped or cancelled. To create this metric:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Metric**.
2. In the next window, for Name, type `Order Status`. For Type, select **String**. Then click **OK**.

Archived

3. In the Monitor Details Model panel (Figure 5-43 on page 193):
   a. Under Metric Details:
      i. Select **A value is required for this metric**.
      ii. For Default Value, type `'New'` with the single quotation marks.
   b. Under Metric Value Expressions:
      i. Click the **Add** button.
      ii. In the first row of the Metric Value Expressions table, select the cell under the Trigger column and click the button in the cell.
      iii. In the Select a Trigger window, select **Trigger type**. Then select **OrderHandling** → **Order Shipped Trigger** and click **OK**.
      iv. In the first row of the Metric Value Expressions table, select the cell under the Expression column and click the button in the cell.
      v. Type `'Shipped'` with the single quotation marks, and click **OK**.
      vi. Click the **Add** button again.
      vii. In the first row of the Metric Value Expressions table, select the cell under the Trigger column and click the button in the cell.
      viii. In the Select a Trigger window, select **Trigger type**. Then select **OrderHandling** → **Order Cancelled Trigger** and click **OK**.
      ix. In the first row of the Metric Value Expressions table, select the cell under the Expression column and click the button in the cell.
      x. Type `'Cancelled'` with the single quotation marks and click **OK**.

Archived

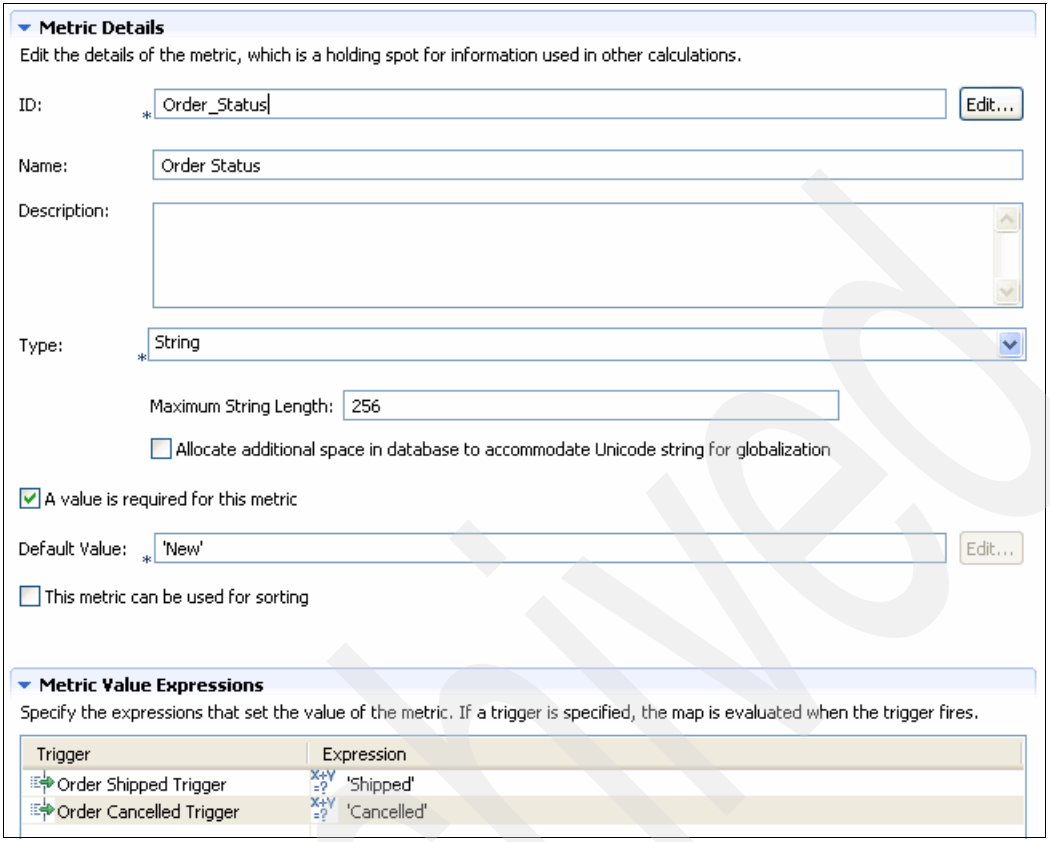


Figure 5-43 Order Status metric overview

4. Press Ctrl+S to save your work.

Total Price

The Total Price metric is used as part of a process instance basic information set. To create this metric:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Metric**.
2. In the next window, for Name, type `Total Price`. For Type, select **Decimal**. Then click **OK**.

Archived

3. In the Monitor Details Model panel, under Metric Value Expressions (Figure 5-43):
   a. Click the **Add** button.
   b. In the first row of the Metric Value Expressions table, select the cell under the Trigger column and click the button in the cell.
   c. In the Select a Trigger window, select **Trigger type**. Then select **OrderHandling** → **Order Received Trigger** and click **OK**.
   d. In the first row of the Metric Value Expressions table, select the cell under the Expression column and click the button in the cell.
   e. Type `Order_Handling_ReceiveEXIT/Input/TotalPrice` and click **OK**.

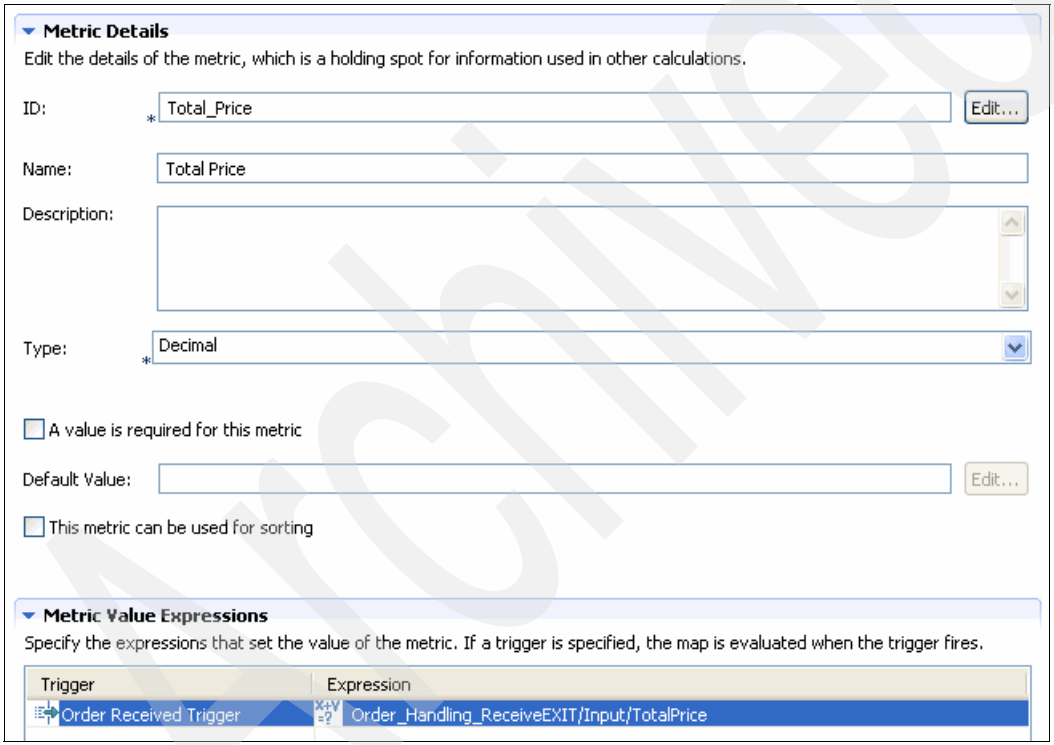


Figure 5-44 Total Price metric overview

4. Press Ctrl+S to save your work.

Archived

Order Shipped

The Order Shipped metric is used by the percentage of shipped orders KPI. This metric either contains the value 0 (Order is cancelled) or 100 (Order is shipped). To create this metric:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Metric**.
2. In the next window, for Name, type `Order Shipped`. For Type, select **Integer**. Then click **OK**.
3. In the Monitor Details Model panel, under Metric Value Expressions (Figure 5-45 on page 196):
   a. Click the **Add** button.
   b. In the first row of the Metric Value Expressions table, select the cell under the Trigger column and click the button in the cell.
   c. In the Select a Trigger window, select **Trigger type**. Then select **OrderHandling** → **Order Cancelled Trigger** and click **OK**.
   d. In the first row of the Metric Value Expressions table, select the cell under the Expression column and click the button in the cell.
   e. Type `0` and click **OK**.
   f. Click the **Add** button again.
   g. In the first row of the Metric Value Expressions table, select the cell under the Trigger column and click the button in the cell.
   h. In the Select a Trigger window, select **Trigger type**. Then select **OrderHandling** → **Order Shipped Trigger** and click **OK**.
   i. In the first row of the Metric Value Expressions table, select the cell under the Expression column and click the button in the cell.
   j. Type `100` and click **OK**.

Archived

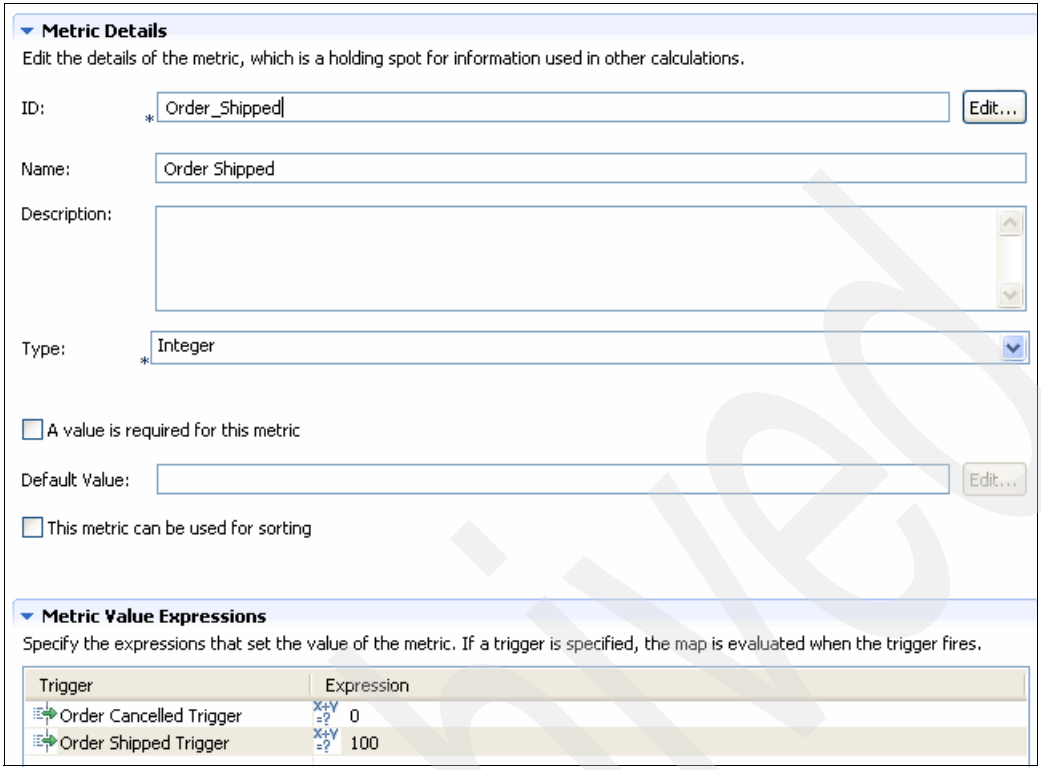


Figure 5-45 Order Shipped metric overview

4. Press Ctrl+S to save your work.

Order Number

The Order Number metric is used by an alerts WebSphere Business Monitor will send at runtime. To create this metric:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Metric**.
2. In the next window, for Name, type `Order Number`. For Type, select **Integer**. Then click **OK**.

Archived

3. In the Monitor Details Model panel, under Metric Value Expressions (Figure 5-46):
   a. Click the **Add** button.
   b. In the first row of the Metric Value Expressions table, select the cell under the Trigger column and click the button in the cell.
   c. In the Select a Trigger window, select **Trigger type**. Then select **OrderHandling** → **Order Received Trigger** and click **OK**.
   d. In the first row of the Metric Value Expressions table, select the cell under the Expression column and click the button in the cell.
   e. Type `Order_Handling_ReceiveEXIT/Input/OrderNumber` and click **OK**.

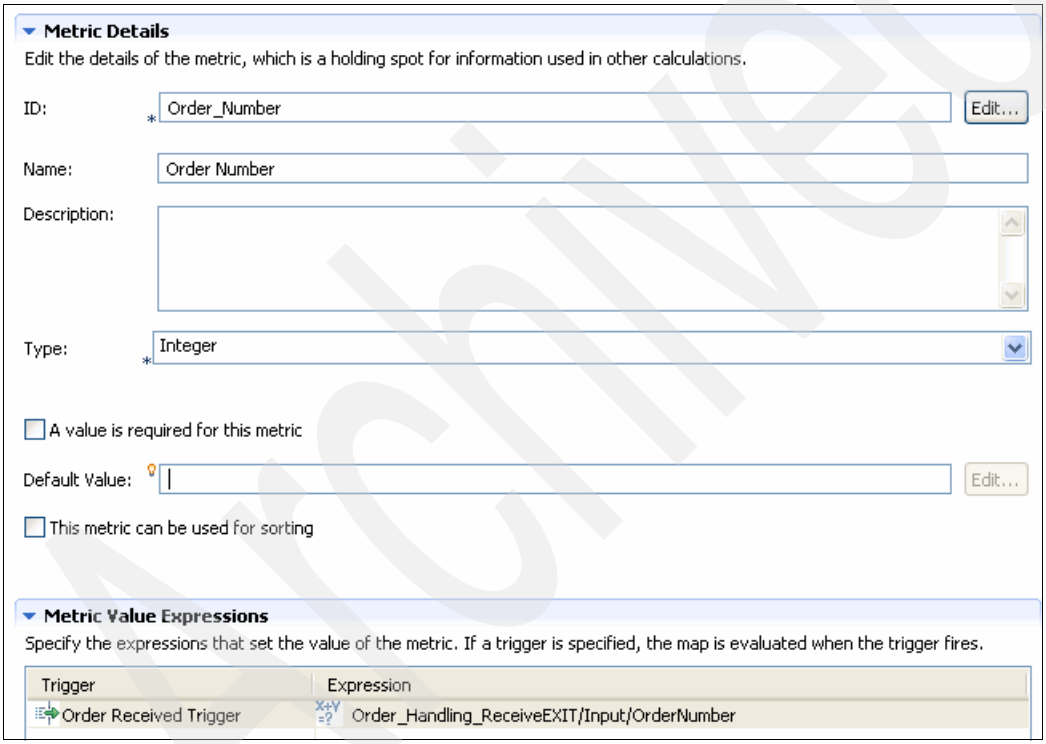


Figure 5-46 Order Number Metric overview

4. Press Ctrl+S to save your work.

Archived

Creating a counter

A counter is used to count all shipped orders. To create a counter:

1. In the model navigation view, right-click **OrderHandling** and select **New** → **Counter**.
2. In the next window, for Name, type `Shipped Order Counter` and click **OK**.
3. In the Monitor Details Model panel, under Counter Controls (Figure 5-47):
   a. Click **Add**.
   b. In the next window, select **OrderHandling** → **Ship OrderEXIT**. Click **OK**.
   c. Under Resulting Action, accept the default of **Add One**.

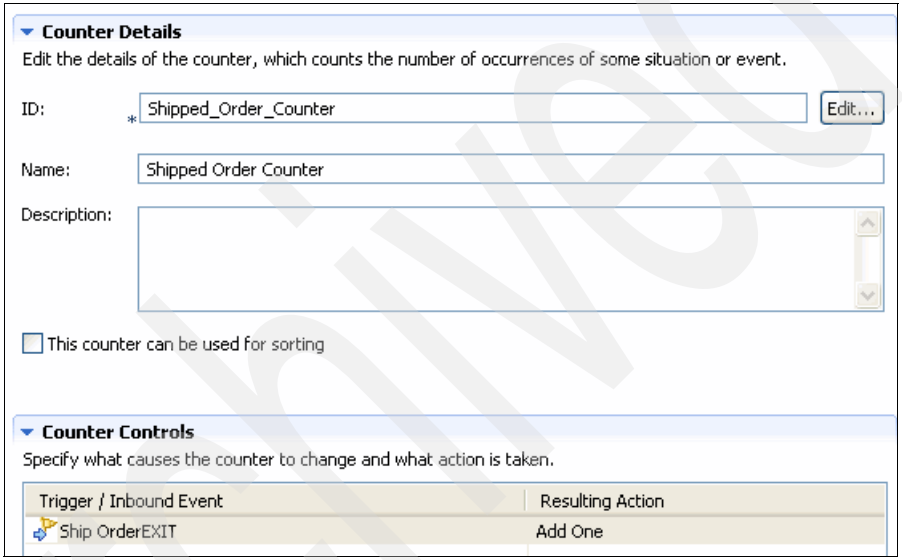


Figure 5-47 Shipped Order Counter overview

4. Press Ctrl+S to save your work.

Creating KPIs

KPIs are defined within a KPI context. In the following sections, we explain how to create KPIs for the monitor model.

Archived

Creating a KPI context

To create a KPI context:

1. Click the **KPI Model** tab for the ClipsAndTacks monitor model.
2. In the KPI Model panel:
   a. Right-click **ClipsAndTacksMM** and select **New** → **KPI Context**.
   b. Under KPI Context Details (Figure 5-48), for Name, type `ClipsAndTacks KC` and click **OK**. A new KPI context is created.
3. Press Ctrl+S to save your work.

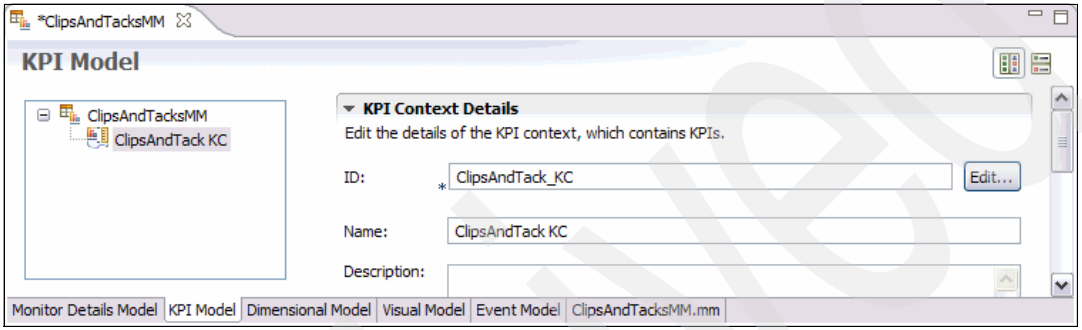


Figure 5-48 ClipsAndTacks KPI context

Average Order Fulfillment

The Average Order Fulfillment KPI indicates the average duration an order needs to be fulfilled, where fulfillment can be a shipment or a cancellation. To create this KPI:

1. In the navigation view of the KPI model, right-click **ClipsAndTacks KC** and select **New** → **KPI**.
2. In the next window, for Name, type Enter `Average Order Fulfillment` and click **OK**.

Archived

3. In the KPI Model panel (Figure 5-49 on page 201):
   a. In the KPI Details section, for Type, select **Duration**.
   b. Under KPI Target and Ranges:
      i. Click the **Details** button. Change the value to `3 minutes` and click **Finish**. Note that three *minutes* are used instead of three *days* to simplify the Dashboard simulation.
      ii. For Ranges, select **Actual value**.
      iii. In the Range table, click **Add**. For Name, type `Day 1-2` and click **OK**.
      iv. A row is created in the Range table. The Start value is set to `0` milliseconds, which is correct.
      v. Select the cell in the End value column and click the button in the cell.
      vi. Change the value to `2 minutes` and click **OK**.
      vii. In the Range table, click **Add** again. For Name, type `Day 3` and click **OK**.
      viii. Set the Start value to `2 minutes` and the End value to `3 minutes`.
      ix. In the Range table, click **Add** again. For Name, type `Day 4-5` and click **OK**.
      x. Set the Start value to `3 minutes` and the End value to `10 minutes`.
   c. Under KPI Value, select **Base this KPI on a metric and an aggregation function**.
   d. Under KPI Details:
      i. For Monitoring context, click **Browse**. Select **ClipsAndTacksMM** → **OrderHandling** and click **OK**.
      ii. For Metric, click **Browse**. Select **OrderHandling** → **Order Fulfillment Duration** and click **OK**.
      iii. For Aggregation function, select **Average**.

Archived

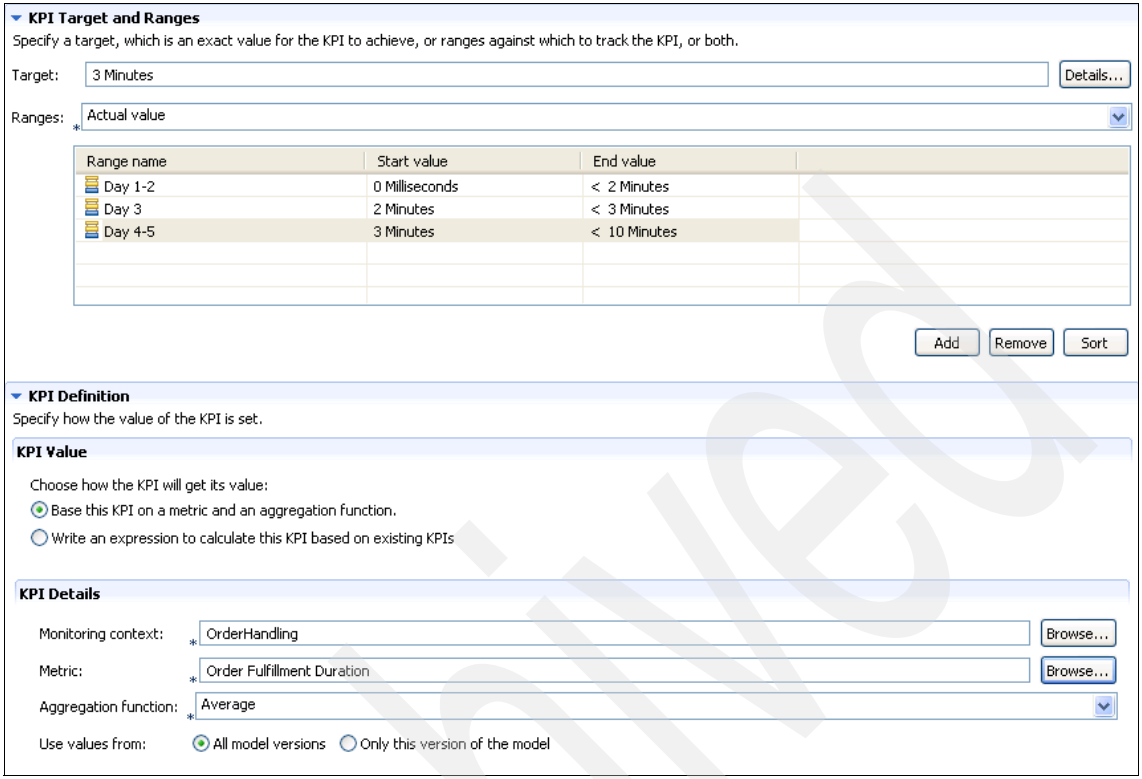


Figure 5-49 Average Order Fulfillment KPI overview

4. Press Ctrl+S to save your work.

Percentage of Shipped Orders

The Percentage of Shipped Orders KPI shows the percentage of shipped orders. To create this KPI:

1. In the navigation view of the KPI model, right-click **ClipsAndTacks KC** and select **New** → **KPI**.
2. In the next window, for Name, type `Percentage of Shipped Orders` and click **OK**.
3. In the KPI Model panel (Figure 5-50 on page 202):
   a. In the KPI Details section, for Type, select **Decimal**.
   b. Under KPI Target and Ranges:
      i. Click the **Details** button. Change the value to `90` and click **Finish**.
      ii. For Ranges select **Actual value**.

Archived

iii. In the Range table, click **Add**. For Name, type `Low Range` and click **OK**.

iv. Set the Start value to `0` and the End value to `90`.

v. In the Range table, click **Add** again. For Name, type `High Range` and click **OK**.

vi. Set the Start value to `90` and the End value to `100`.

c. Under KPI Value select **Base this KPI on a metric and an aggregation function**.

d. Under KPI Details:

i. For Monitoring context, click **Browse**. Select **ClipsAndTacksMM** → **OrderHandling** and click **OK**.

ii. For Metric, click **Browse**. Select **OrderHandling** → **Order Shipped** and click **OK**.

iii. For Aggregation function, select **Average**.

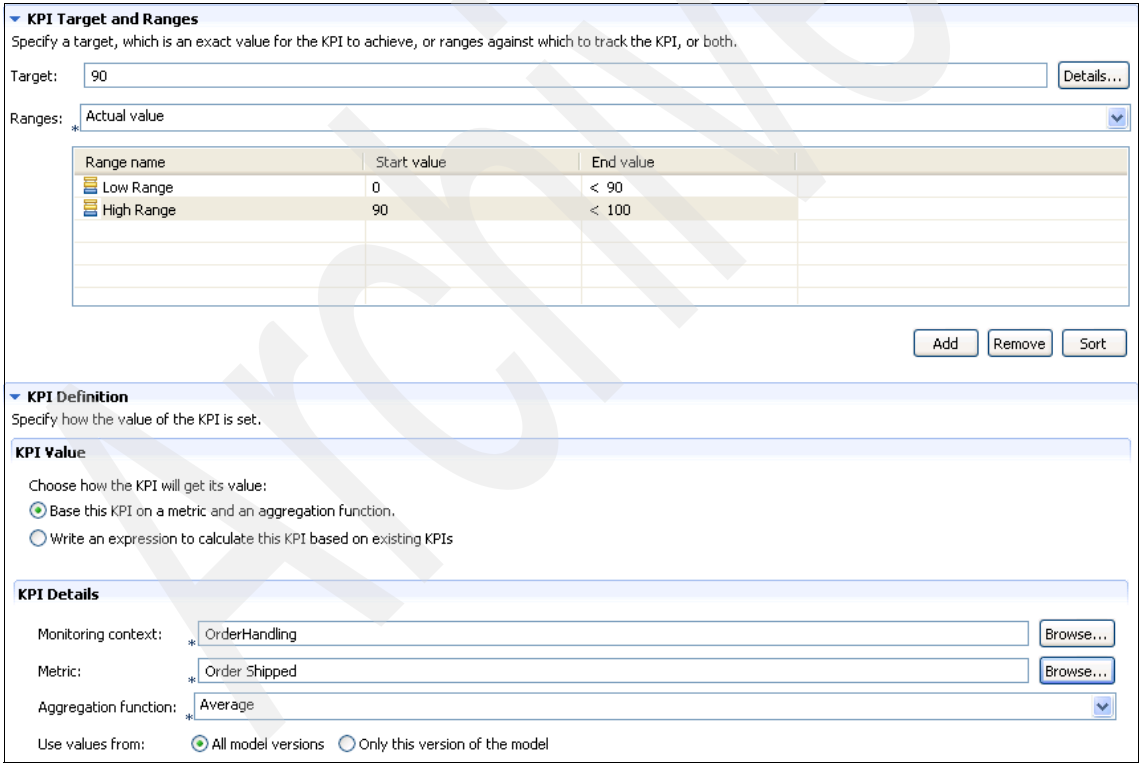


Figure 5-50 Percentage of Shipped Orders KPI overview

4. Press Ctrl+S to save your work.

Archived

Creating a dimension

WebSphere Business Monitor offers a dimension view that enables drill downs to lower dimensions based on metrics and measures. In this section, we explain how to create one dimension for the location to which an order is shipped and one measurement.

Location

To create a dimension for the location:

1. Click the **Dimensional Model** tab.
2. In the navigation view, right-click **OrderHandling Cube** and select **New** → **Dimension**.
3. For Name, type `Location` and click **OK**.
4. In the navigation view, right-click **Location** and click **New** → **Dimension Level**.
5. In the right pane:
   a. For Name, type `Country`.
   b. For Source metric, click **Browse**. Select **OrderHandling** → **Country**. Click **OK**.
   c. Click **OK**.
6. In the navigation view, right-click **Location** and click **New** → **Dimension Level**.
7. In the right pane:
   a. For Name, type `City`.
   b. For Source metric, click **Browse**. Select **OrderHandling** → **City**. Click **OK**.
   c. Click **OK**.

   As shown in the Figure 5-51 a new dimension is created with two dimension levels, Country and City.

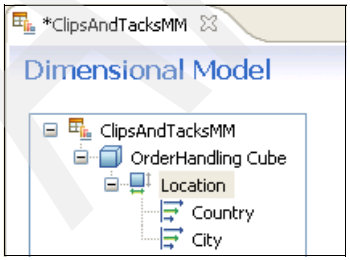


Figure 5-51 Location dimension

8. Press Ctrl+S to save your work.

Archived

Measurement

To develop a measure for shipped orders:

1. In the navigation view, right-click **OrderHandling Cube** and click **New** → **Measure**.
2. In the Create New Measure window (Figure 5-52 on page 204):
   a. For Name, type `Number of Shipped Orders`.
   b. For Source metric, click **Browse**. Select **OrderHandling** → **Shipped Order Counter**. Click **OK**.
   c. For Aggregation function, select **Sum**.
   d. Click **OK**.

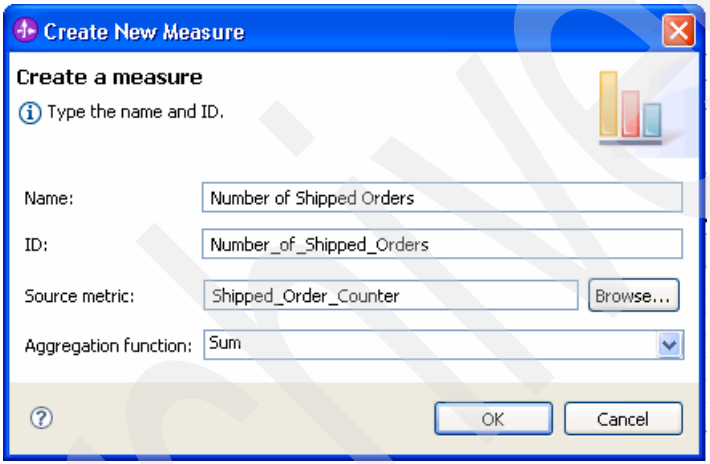


Figure 5-52 Number of Shipped Orders Measure

3. Press Ctrl+S to save your work.

Creating an alert

ClipsAndTacks wants to receive alerts on a dashboard whenever an order takes longer than three days. Alerts are realized by creating outbound events within the monitor model that are handled by WebSphere Business Monitor at run time. The event is configured in this section, but server settings to handle that alert at run time are defined later. For a unique alert message in the dashboard, the event must contain information about the order instance. The alert is based on the Late Order Fulfillment Event Trigger defined earlier.

Creating a Late Order Shipped Event

To create a Late Order Shipped Event:

1. In the Project Explorer view, right-click **Event Definitions** and select **New** → **Event Definition...(cbe)**.
2. In the New Event Definition window (Figure 5-53 on page 205), select the **ClipsAndTacksMonitoring** folder. For File name, type `LateOrderShippedEvent.cbe`. Click **Finish**.

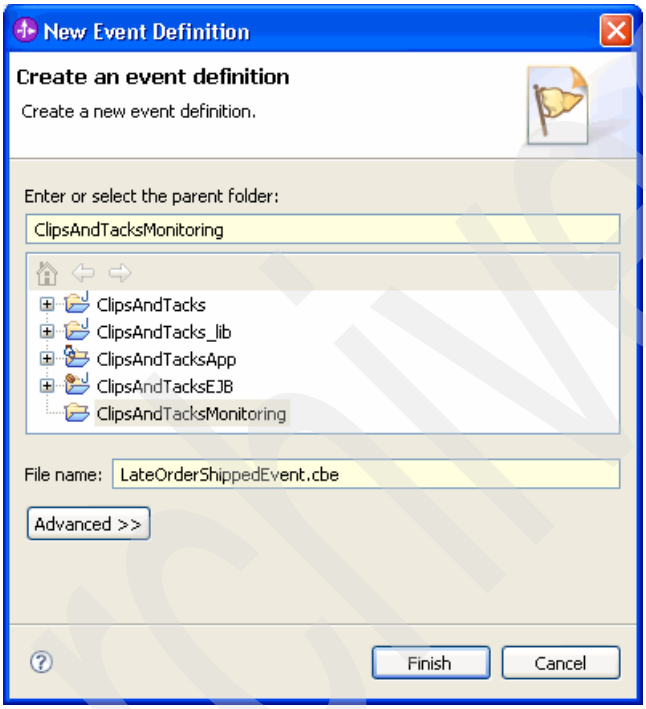


Figure 5-53 Create the Late Order Shipped Event

3. The Event Definition Editor for LateOrderShippedEvent opens. Double-click the **hierarchical** icon ( ) next to the Parent label.

Archived

4. In the Select Event Definition window (Figure 5-54), double-click **ActionServicesEvent**.

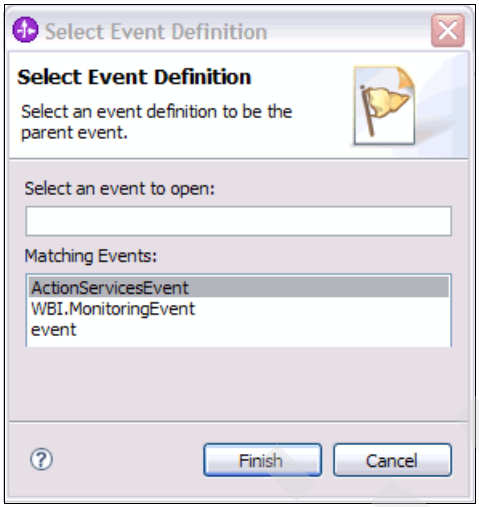


Figure 5-54 Select Event Definition window

ActionServicesEvent is now shown as the parent of LateOrderShippedEvent.

5. On the event definition, click the **Add Extended Data** icon ( ) to adds a row to the Extended Data table.
6. Enter the following values for the new data element:
   a. For Extended Data name, type `OrderProcessingTime`.
   b. For Type, select **string**.
7. On the event definition, click the **Add Extended Data** icon again.
8. Enter the following values for the new data element:
   a. For Extended Data Name, type `OrderNumber`.
   b. For Type, select **string**.

Archived

The LateOrderShippedEvent view shows two extended Data properties (Figure 5-55).

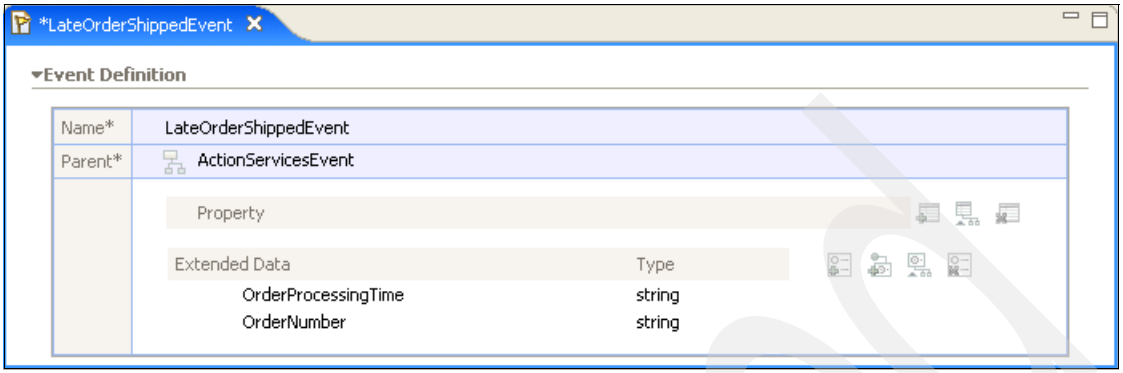


Figure 5-55 LateOrderShippedEvent overview

9. Press Ctrl+S to save your work.

Creating an Outbound Event

To create an Outbound Event:

1. Click the **Monitor Details Model** tab of the ClipsAndTacksMM monitor model.
2. In the navigation view, right-click **OrderHandling** and select **New** → **Outbound Event**.

Archived

3. In the Create New Outbound Event window (Figure 5-56):
   a. For Name, type `Late Order Shipped Event`.
   b. Select **Configure this event to be processed by WebSphere Business Monitor action services**.
   c. For Trigger, click **Browse**. Select **OrderHandling** → **Late Order Fulfillment Event**. Click **OK**.
   d. Click **OK**.

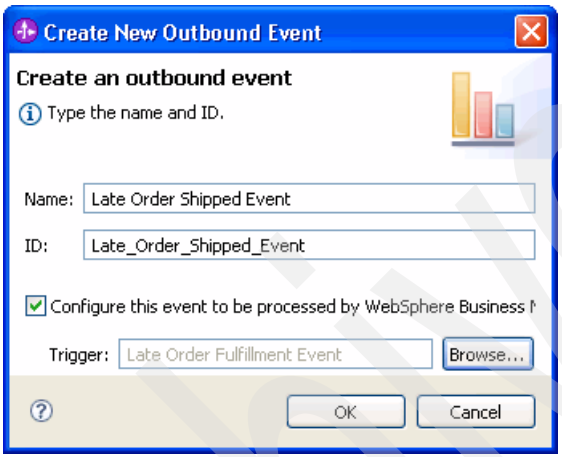


Figure 5-56 Creating the Late Order Shipped Event Outbound Event

Archived

4. In the Monitor Details Model panel (Figure 5-57):
   a. Under Event Type Details, for Extension name, click **Browse**. Select **ClipsAndTacksMonitoring** → **LateOrderShippedEvent.cbe** → **LateOrderShippedEvent** and click **OK**.
   b. Under Event Attributes Details, in the table, expand **Late Order Fulfillment Event** → **Extended Data** and type the following values:
      i. For OrderProcessingTime, type `xs:string(Order_Fulfillment_Duration)`.
      ii. For OrderNumber, type `xs:string(Order_Number)`.
      iii. For BusinessSituationName, type `'Order handling duration is too long'` with the quotation marks.

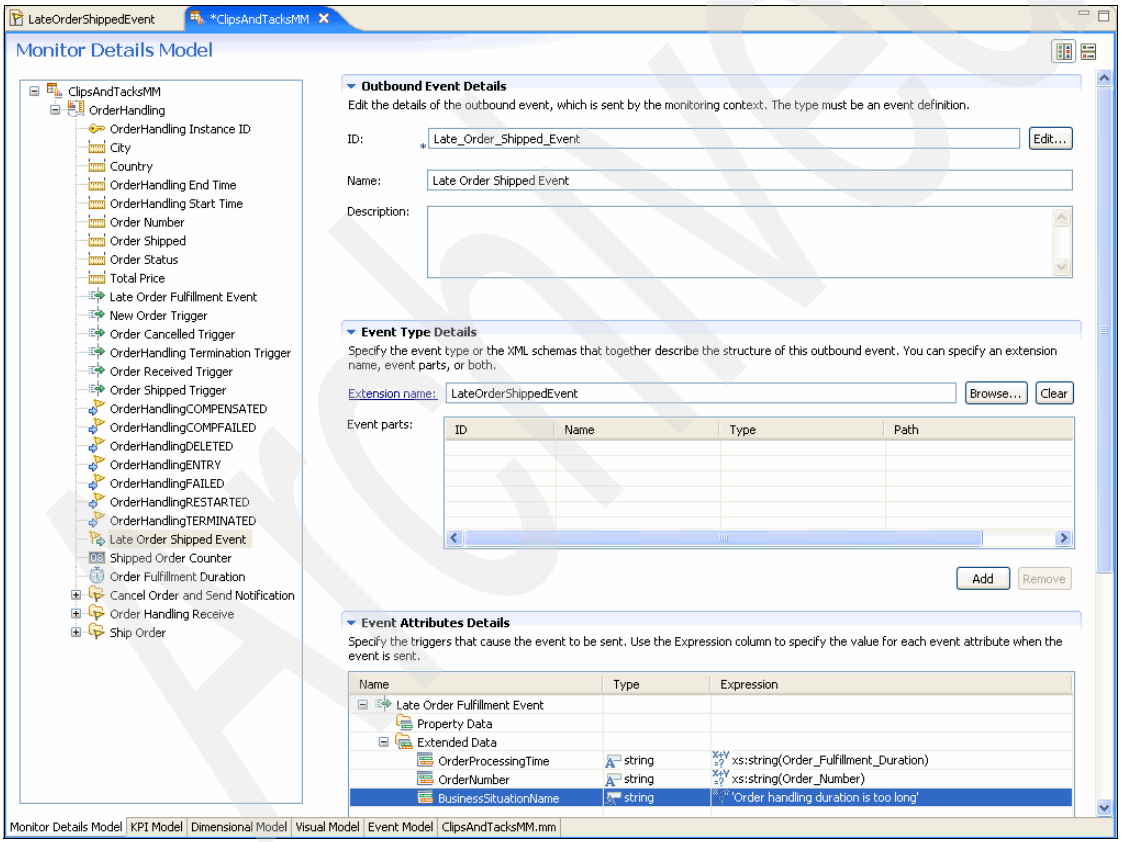


Figure 5-57 Late Order Shipped Event Outbound Event overview

5. Press Ctrl+S to save your work.

Creating visual models

ClipsAndTacks has a visual representation of the Order Handling process in terms of a Scalable Vector Graphics (SVG) file. They want to use this image to visualize whether an order has been shipped or cancelled and to have an overview about the percentage of the shipped orders KPI. To display that in a WebSphere Business Monitor dashboard, you must configure visual models.

To import the two supplied SVG diagrams:

Note: The Order_Handling_MC.svg and Order_Handling_KC.svg files are provided in the additional material supplied with this book. See Appendix B, “Additional material” on page 801, for details.

1. Right-click in the Project Explorer view and select **Import**.
2. In the Import window, expand **General** and select **File system**.Click **Next**.
3. In the Import – File System window (Figure 5-58):
   a. For From directory, click **Browse** to locate the directory that contains the two SVG files. Select both SVG files.
   b. For Into folder, click the **Browse** button. In the Import Into Folder window, select **ClipsAndTacksMonitoring** and click **OK**.
   c. Click **Finish**.

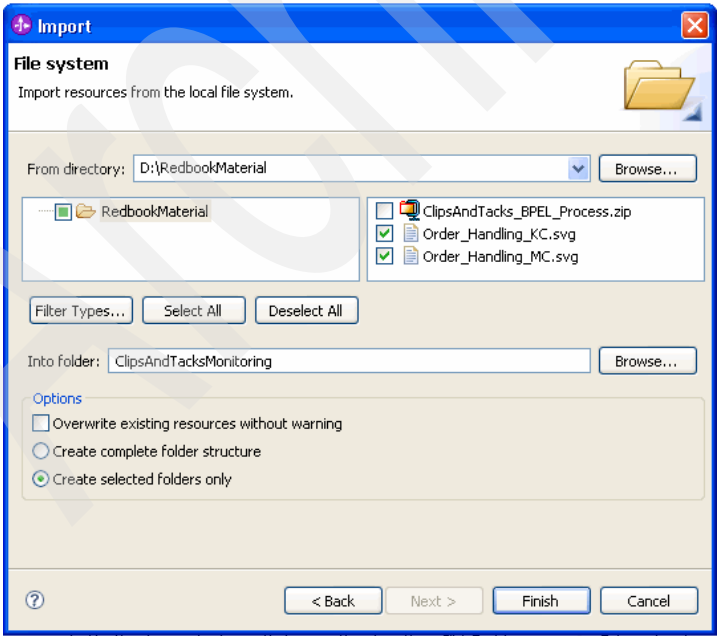


Figure 5-58 Importing the ClipsAndTacks SVG files

Creating a visual model for monitoring context

When creating the visual model for the monitoring context, we set up the model so that the Ship Order activity changes to green when an order is shipped. A Cancel Order And Sent Notification activity changes to red when an order is cancelled. To create the visual model for monitoring context:

1. At the bottom of the editor view, click the **Visual Model** tab.
2. The Selected Context defaults to the monitoring context **OrderHandling**, which is correct.
3. To choose an SVG file, click **Browse**. In the Select SVG Resource window (Figure 5-59), expand **ClipsAndTacksMonitoring** and select **Order_Handling_MC.svg** and click **OK**.

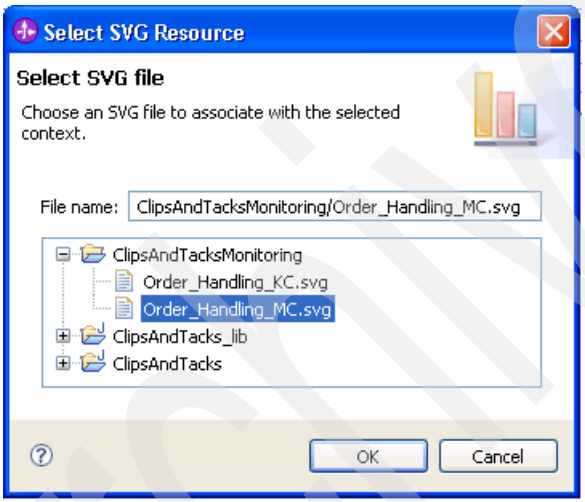


Figure 5-59 Order Handling MC SVG file selection

4. In the message window that prompts whether to create shape sets based on SVG files, click **Yes** to create shapes for each component of the SVG diagram.

Archive

The SVG diagram opens in the visual editor as shown in Figure 5-60.

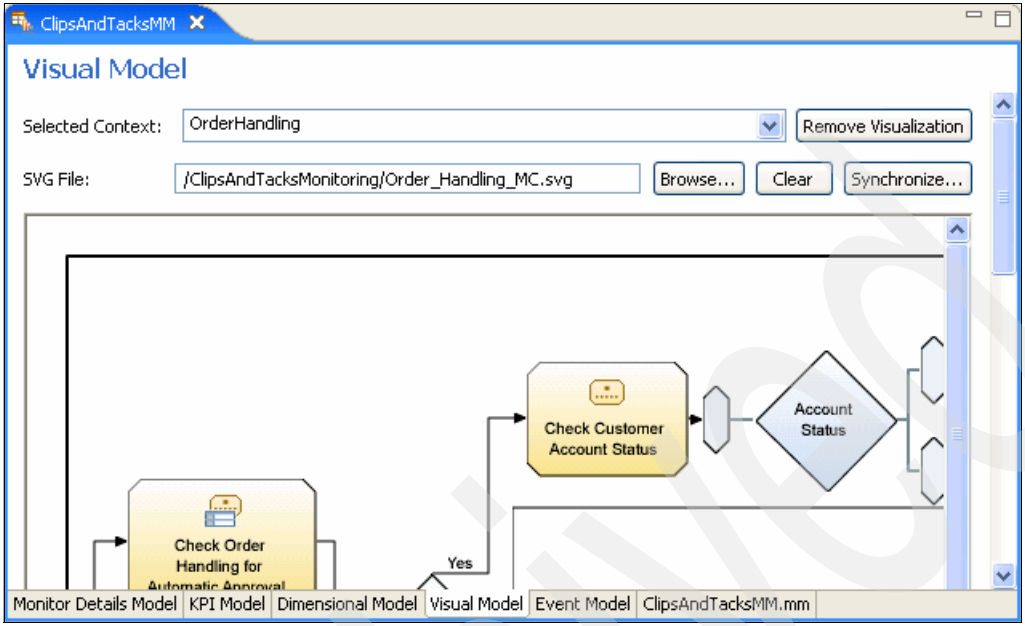


Figure 5-60 Imported Order Handling MC SVG file

5. Scroll to the right in the diagram to the tasks *Ship Order* and *Cancel Order and Send Notification*, which are the tasks to be manipulated.
6. Based on the metric Order Status, set the color of the task *Ship Order* to green to indicate that an order has shipped:
   a. The bottom left portion of the visual editor view the Add and Edit Shape set Actions section, which contains all available SVG shapes. Under Shape Set / Action, right-click **Ship_Order** and select **Add Action** → **Set Color** (Figure 5-61).

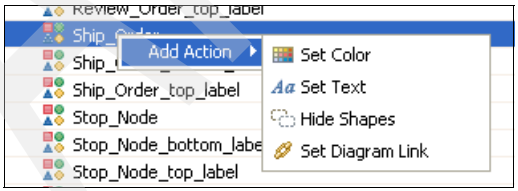


Figure 5-61 Selecting the shape action

b. Enter the following values:

i. Click in the cell to the right of *condition*. Click the button in the expression field, type `Order_Status eq 'Shipped'`, and click **OK**.

ii. Click in the cell to the right of *fill color*. Click the button in the field, select **green**, and click **OK**.

Figure 5-62 shows the result of these changes.

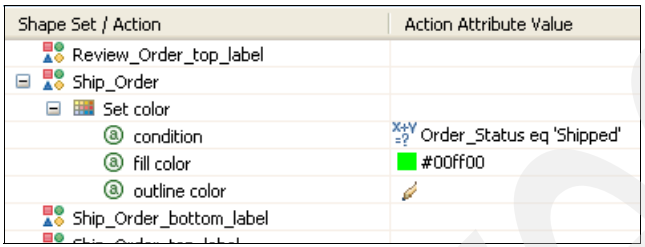


Figure 5-62 Setting the fill color for Ship Order

7. Based on the metric Order Status, set the color of the task *Cancel Order and Send Notification* to red to indicate that an order has been cancelled:

a. Under Shape Set / Action, right-click **Cancel_Order_and_Send_Notification** and select **Add Action → Set Color**.

b. Enter the following values:

i. Click in the cell to the right of *condition*. Click the button in the expression field, type `Order_Status eq 'Cancelled'`, and click **OK**.

ii. Click in the cell to the right of *fill color*. Click the button in the field, select **red**, and click **OK**.

Figure 5-63 shows the result of these changes.

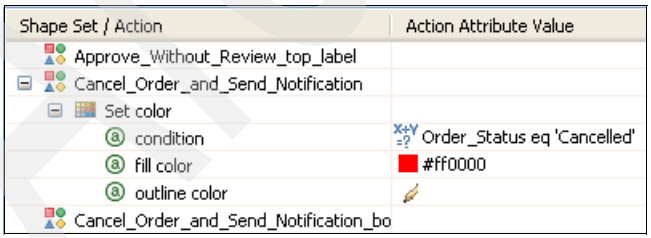


Figure 5-63 Setting the fill color for Cancel Order and Send Notification

8. Press Ctrl+S to save your work.

Archived

9. Test the settings by using the diagram test facility:

   a. The lower right corner of the visual editor view shows all metrics listed that are referenced in the visual model. Values can be set to see how the model will behave. For the metric Order Status, type `Shipped` (Figure 5-64).

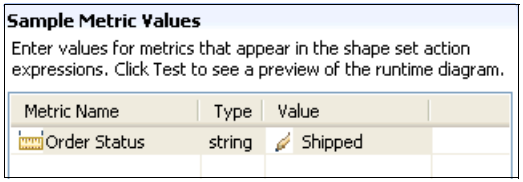


Figure 5-64 Entering a sample metric value

   b. Click the **Test** button.

   c. In the diagram preview window (Figure 5-65), scroll to the right to check that the task *Ship Order* has changed to green. Close the diagram preview window.

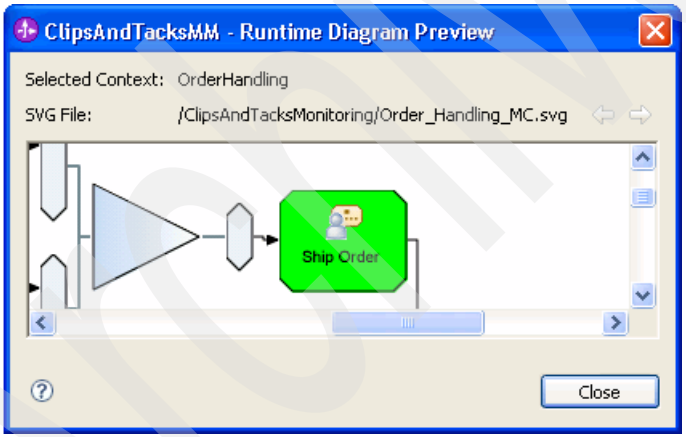


Figure 5-65 Testing the monitoring context visual model

Creating a visual model for KPI context

In this section we define the visual model for the KPI context. The percentage of orders that have a status of *Shipped* should be displayed in the diagram. The Ship Order activity should change to red when the value of this KPI goes below 90%. Otherwise, this activity should change to green. To create the visual model for KPI context:

1. At the bottom of the editor view, click the **Visual Model** tab. Set Selected Context to **ClipsAndTacks KC**.

Archived

2. To choose an SVG file, click **Browse**. In the SVG Resource Selection window, expand **ClipsAndTacksMonitoring** and select **Order_Handling_KC.svg** and click **OK**.
3. When you are prompted to create shape sets based on SVG files, click **Yes**.
4. The SVG diagram opens in the visual editor.
5. Based on the KPI Percentage_of_Shipped_Orders, the color of the Ship Order shape must be set to green when the percentage of shipped orders is higher than 90%. Otherwise, the color must be set to red (Figure 5-66).
   a. The bottom left portion of the visual editor view shows *Add and Edit Shape set Actions*. Under Shape Set / Action, right-click **Ship_Order_2** and select **Add Action** → **Set Color**.
   b. Enter the following values:
      i. Click in the cell to the right of *condition*. Click the button in the expression field, type `Percentage_of_Shipped_Orders ge 90`, and click **OK**.
      ii. Click in the cell to the right of *fill color*. Click the button, select **green**, and click **OK**.
   c. Right-click the **Ship_Order_2** item again and select **Add Action** → **Set Color**.
   d. Enter the following values:
      i. Click in the cell to the right of *condition*. Click the button in the expression field, type `Percentage_of_Shipped_Orders lt 90`, and click **OK**.
      ii. Click in the cell to the right of *fill color*. Click the button, select **red**, and click **OK**.

      Figure 5-66 shows the completed values.

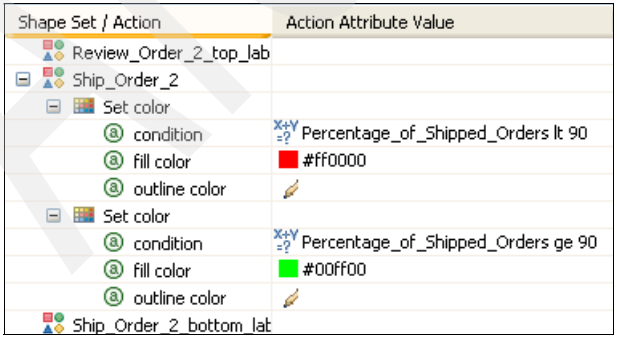


Figure 5-66 Setting the Ship Order fill color KPI

Archive

6. Set the label below the *Ship Order* activity to the current value of the Percentage_of_Shipped_Orders KPI (Figure 5-67):
   a. The bottom left portion of the visual editor view shows *Add and Edit Shape set Actions*. Under Shape Set / Action, right-click **Ship_Order_2_bottom_label** and select **Add Action** → **Set Text**.
   b. Click in the cell to the right of *text value*. Click the button in the expression field and type `fn:concat('Percentage of shipped orders: ',Percentage_of_Shipped_Orders, '%')`. Then click **OK**.

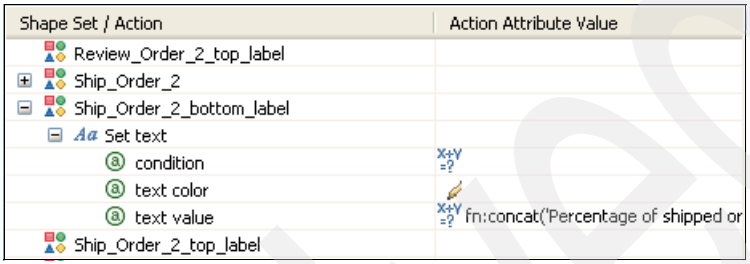


Figure 5-67 Setting the Ship Order Label KPI

7. Press Ctrl+S to save your work.
8. Test the settings by using the diagram test facility (Figure 5-68).

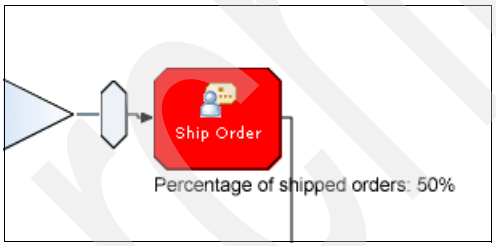


Figure 5-68 Testing the KPI visual model

5.3.6 Testing the integration

In this section, we test the ClipsAndTacks application and the monitor model that we developed in the previous section. We include information about the deployment, server setup, and dashboard creation.

Deploying the Order Handling process

We deploy the Order Handling process on a WebSphere Process Server by using WebSphere Integration Developer. In this case, WebSphere Process Server is integrated into the server view of WebSphere Integration Developer. If

Archived

WebSphere Process Server is not integrated into the server view of WebSphere Integration Developer, refer to the “Creating servers in the test environment” topic in the WebSphere Integration Developer Information Center at the following address:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wbit.610.help.runtime.doc/topics/tcreate.html

To deploy the order handling business process on a WebSphere Process Server:

1. Switch to the **Business Integration** perspective.
2. In the Servers view, right-click **WebSphere Process Server v6.1** and select **Add and Remove Projects** (Figure 5-69).

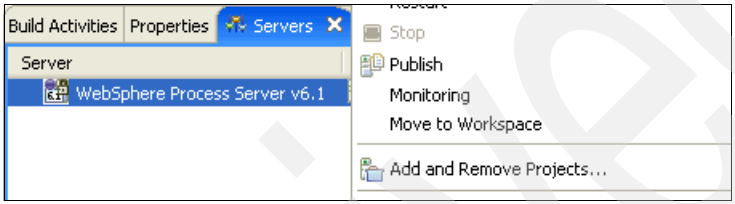


Figure 5-69 Selecting the Add and Remove Projects options

3. In the Add and Remove Projects window (Figure 5-70), select the **ClipsAndTacksApp** project, click **Add**, and then click **Finish**.

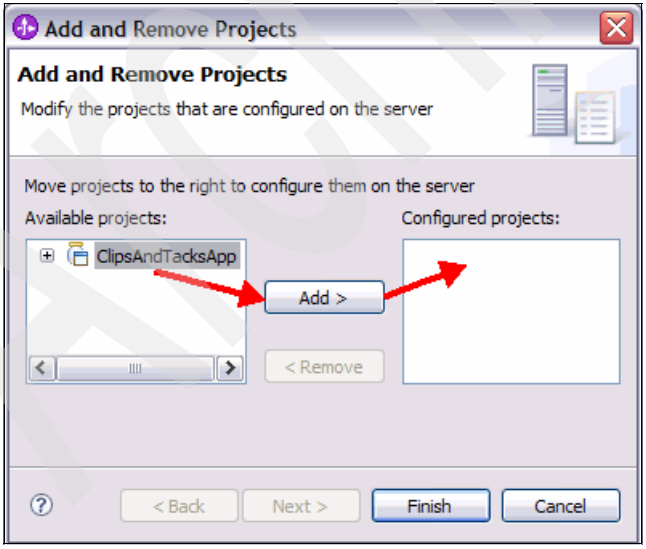


Figure 5-70 Adding the ClipsAndTacks project to WebSphere Process Server

The application is being deployed. Watch the progress indicator in the bottom right of the WebSphere Integration Developer window. After a successful deployment, the application is shown on the Servers view (Figure 5-71).

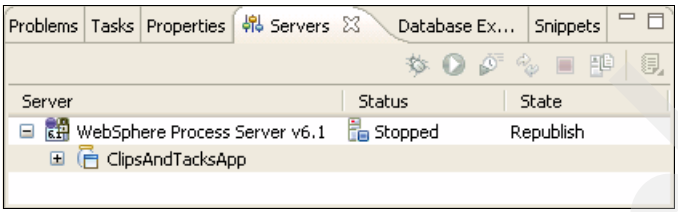


Figure 5-71 ClipsAndTacksApp deployed on WebSphere Process Server

4. Right-click **WebSphere Process Server** and select **Start**.

On the Servers view (Figure 5-72), both the server and the application should have a status of *Started* after a successful start.

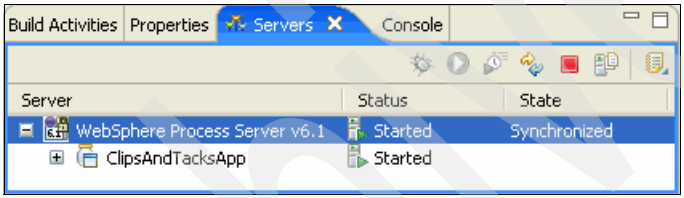


Figure 5-72 WebSphere Process Server with a status of Started

Configuring a remote CEI for WebSphere Business Monitor

The two applications are installed on two different WebSphere Application Server servers in separate cells (Figure 5-73 on page 219). WebSphere Business Monitor contains a Monitor bus and is monitoring events arriving at this bus.

Archived

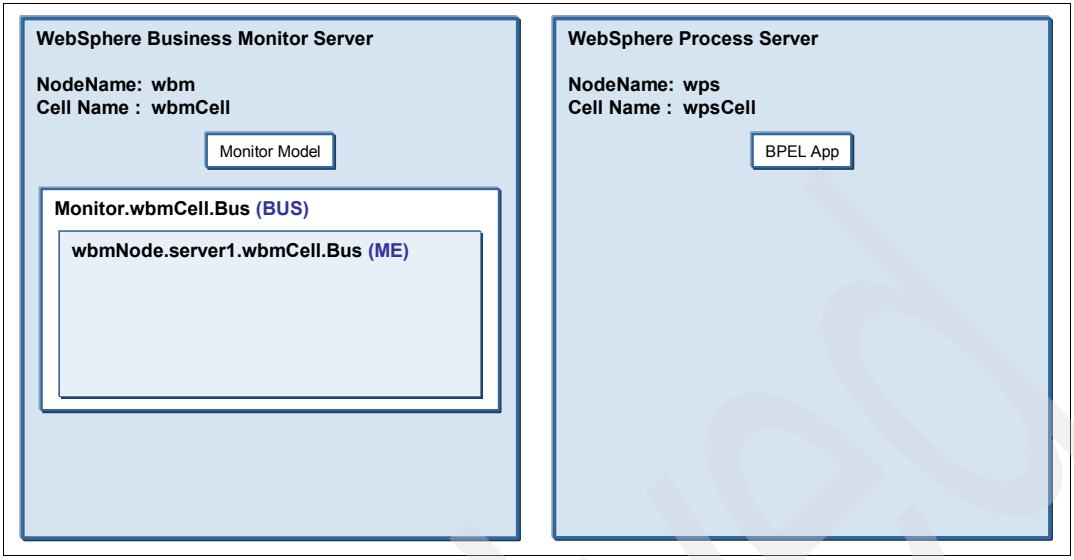


Figure 5-73 WebSphere Business Monitor Monitor bus

To monitor events from WebSphere Process Server, a remote CEI bus must be configured and linked to the WebSphere Business Monitor bus. WebSphere Business Monitor provides a script that creates the remote CEI bus and the service integration bus link as shown in Figure 5-74.

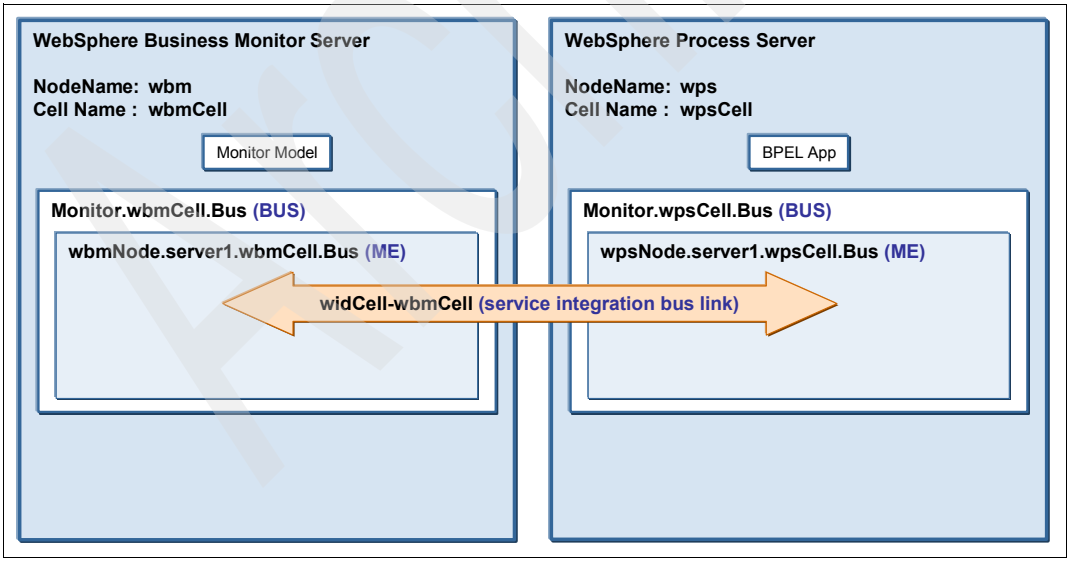


Figure 5-74 Service integration bus link

To configure the remote CEI bus:

1. Ensure that WebSphere Process Server and the WebSphere Business Monitor server are started.
2. From the was_root/scripts.wbm/crossCell folder of the WebSphere Business Monitor server installation, copy the **mon61to61CrossCell.zip** file to the wps_root/plugins folder of the WebSphere Process Server.

   The mon61to61CrossCell.zip extracted files: If the WebSphere Application Server that hosts WebSphere Process Server is also hosting a WebSphere Business Monitor, the files in the mon61to61CrossCell.zip file already exist and must be overwritten to make this scenario work.

3. Extract the contents into the wps_root/plugins folder.
4. From the WebSphere Process Server wps_root/bin folder, run the **osgiCfgInit.bat** or **osgiCfgInit.sh** command. By using this command, WebSphere Process Server recognizes the new plug-in JAR files that were extracted into the plugins directory.
5. From the was_root\scripts.wbm\crosscell folder of the local WebSphere Business Monitor server 6.1 installation, run the service integration bus crosscell configuration utility **configRemoteMonitorBus.bat** or **configRemoteMonitorBus.sh**:
   a. When prompted by the utility, enter the following values:
      i. For Security enabled?, type no.
      ii. For Fully qualified host name of WebSphere Business Monitor, type <host name.DNSsuffix> of WebSphere Business Monitor.
      iii. For SOAP port of WebSphere Business Monitor, check the WebSphere Business Monitor administrative console for the SOAP_CONNECTOR_ADDRESS port. The default port is 8880.

      Note: To access the administrative console of a WebSphere Business Monitor server, type the following URL:

      http://<host name>:<WC_adminhost port>/ibm/console

      iv. For Fully qualified host name of WebSphere Process Server, type <host name.DNSsuffix> of WebSphere Process Server.

v. For SOAP port of WebSphere Process Server, check the WebSphere Process Server administrative console for the SOAP_CONNECTOR_ADDRESS port. The default port is 8880.

vi. For Default configuration for Messaging engines?, type yes.

b. When the script completes, restart both servers.

6. Verify that the remote service integration bus exists and that the link between the local and remote buses created successfully. Log into the administrative console of WebSphere Process Server, expand **Service Integration** and click **Buses**. You see a bus that called Monitor.<cell-name>.bus is created (Figure 5-75).

7. Click **MONITOR.<cell_name.>bus**.

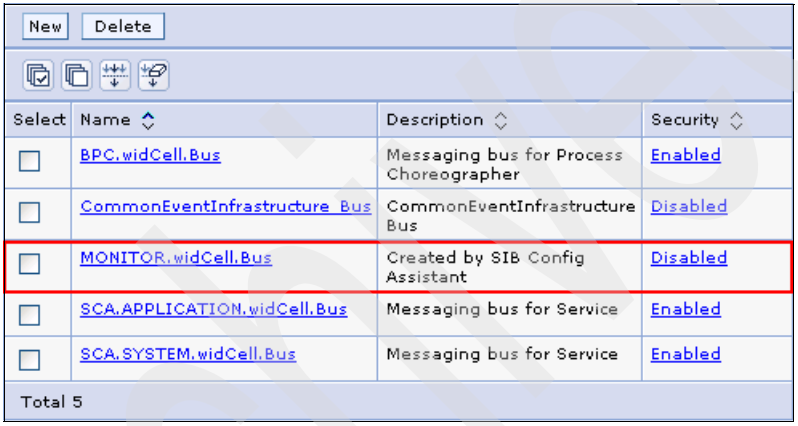


Figure 5-75 Checking the monitor bus creation

8. On the Configuration page (Figure 5-76), under **Topology**, click the **Messaging engines** link.

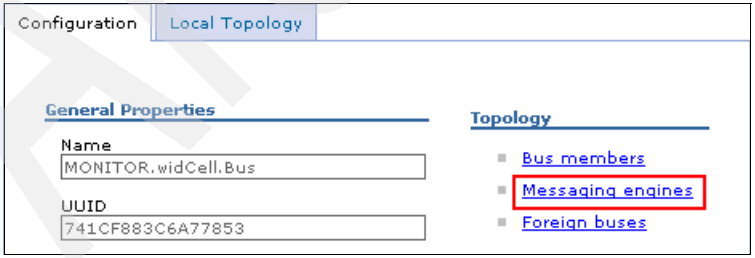


Figure 5-76 Navigating to the monitor bus messaging engines

9. One messaging engine is defined, and the Status field shows a green arrow indicating that the messaging engine is active (Figure 5-77). Click the messaging engine.

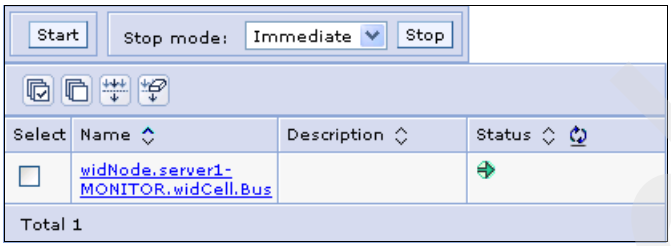


Figure 5-77 Checking the Monitor bus message engine

10. Under Additional Properties, click **Service integration bus links**. One link is defined, and the Status field shows a green arrow indicating that the link is active (Figure 5-78).

Figure 5-78 Checking the service integration bus link

11. Perform the same procedure in step 6 on page 221 through step 10 on the WebSphere Business Monitor for the bus called Monitor.<cell-name>.bus to validate that the service integration bus link is active.

For more information about setting up a remote CEI, see “Configuring a remote CEI server to use WebSphere Business Monitor” in the WebSphere Business Monitor information center:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.btools.help.monitor61.doc/admin/cei_cfg_for_rem_monitor.html`

Deploying the monitor model

To deploy a monitor model, create the necessary J2EE artifacts by using WebSphere Integration Developer:

1. Switch to the **Business Monitoring** perspective.
2. In the Project Explorer view, expand **ClipsAndTacksMonitoring** → **Monitor models**.
3. Right-click **ClipsAndTacksMM.mm** and select **Generate Monitor J2EE Projects** (Figure 5-79).

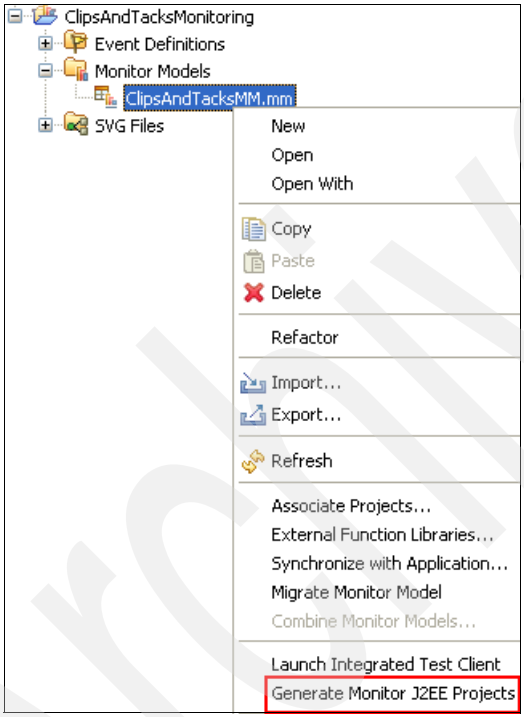


Figure 5-79 Selecting the Generate Monitor J2EE Projects option

Archived

4. In the Generate Monitor J2EE Projects window (Figure 5-80), leave the default values and click **Finish**.

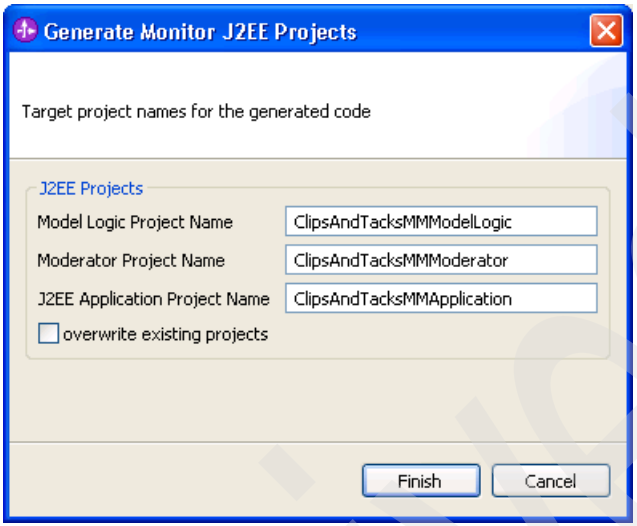


Figure 5-80 Generate Monitor J2EE Projects window

5. Switch to the **Business Integration perspective**.
6. In the Project Explorer, right-click **ClipsAndTacksMMApplication** and select **Export** → **EARFile**.
7. In the Export window (Figure 5-81), for Destination, click **Browse**. Select a destination for the EAR file and click **Save**. Click **Finish** to export the EAR file.

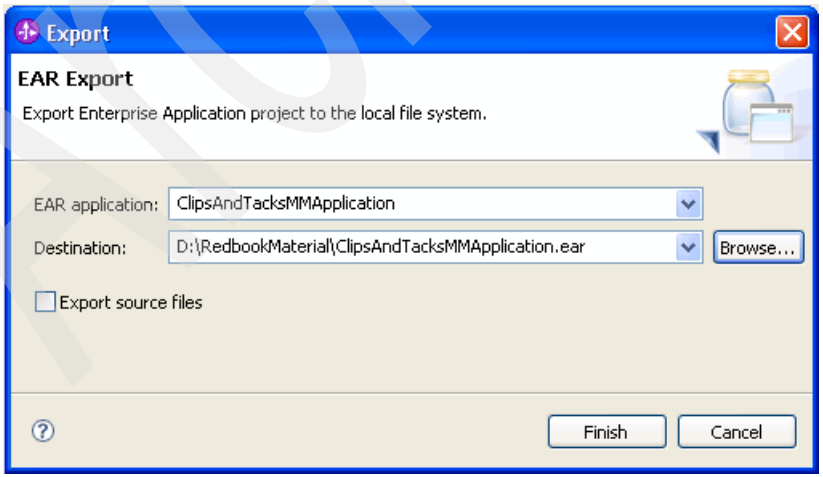


Figure 5-81 Exporting the EAR file

8. To deploy the exported EAR file onto the WebSphere Business Monitor, refer to 3.4.2, “Installing a monitor model by using the remote CEI” on page 97.
9. Start the monitor model as explained in 3.4.3, “Starting and stopping a monitor model” on page 103.

The deployment of the monitor model by using the remote CEI option created a queue and an corresponding queue point on WebSphere Business Monitor server for this monitor model. This queue receives events that are handled by WebSphere Business Monitor. Furthermore a foreign queue is created on WebSphere Process Server that points to the WebSphere Business Monitor queue points. WebSphere Process Server events are sent to the WebSphere Process Server foreign queue. From there, they are sent to the WebSphere Business Monitor queue by using the earlier created Service Integration bus link (Figure 5-82).

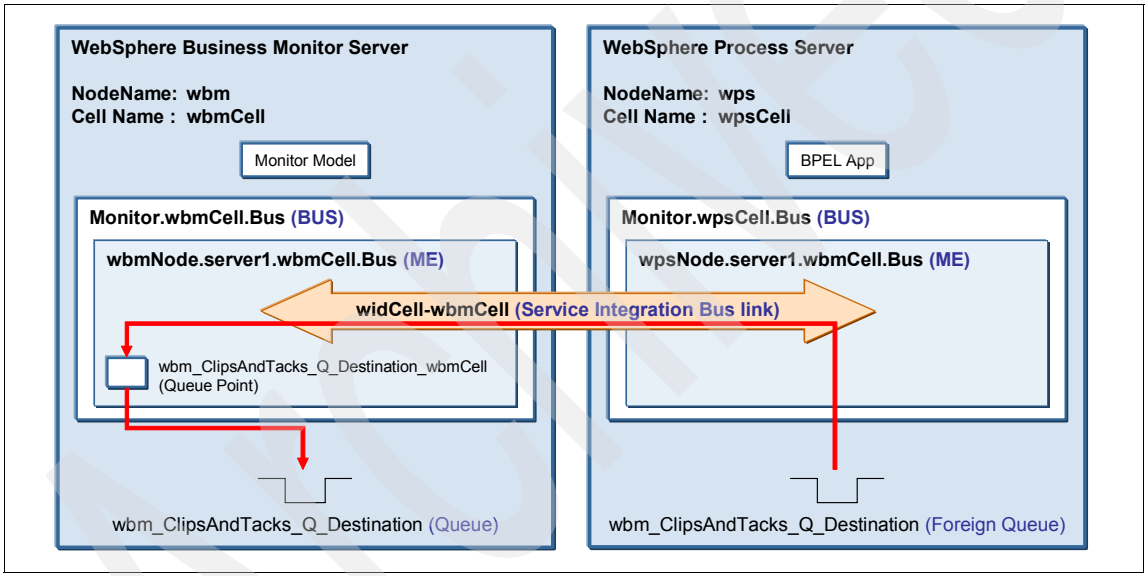


Figure 5-82 Queues created for remote CEI communication

5.3.7 Configuring the action manager for business situation events

The action manager is used to create alerts based on business situations that are defined in outbound events in the monitor model. They are handled by WebSphere Business Monitor at run time and will be visible in the Alert view on the dashboard. In this section, we create an alert for the situation where an order handling process takes longer than three days:

1. Log in to the administrative console of WebSphere Business Monitor.
2. Create a template for late order shipped alerts:
   a. In the WebSphere Business Monitor administrative console (Figure 5-83), expand **Applications** → **Monitor Action Services** → **Template Definition** and click **Notifications**.
   b. In the Notifications panel, click **New**.

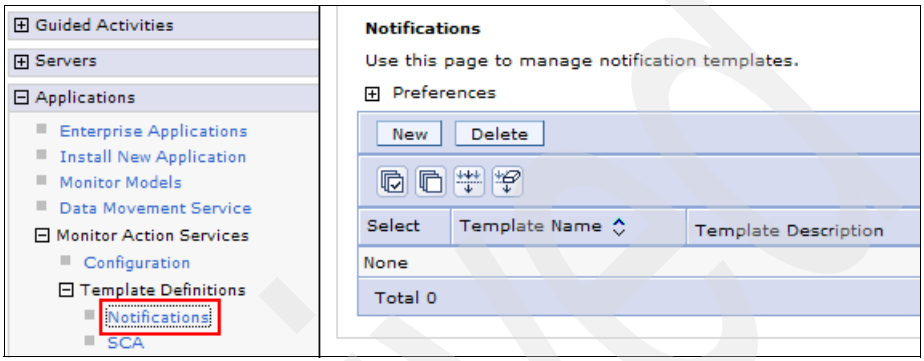


Figure 5-83 Navigating to Notifications

   c. Under Notification Template Configuration (Figure 5-84 on page 227):
      i. For Template name, type `LateOrderShipmentAlert`.
      ii. For Description, type `Late Order Shipment`.
      iii. Under Default action service type, select **Dashboard Alert**.
      iv. Under 'To' query type, select **User id**.
      v. For To, type `admin`. This user ID receives the alert.
      vi. For Subject, type `Late order shipment`.
      vii. For Body, type `The processing time of the order with order number %OrderNumber% is taking too long: %OrderProcessingTime% minutes.`
      viii. Click **OK**.

Archive

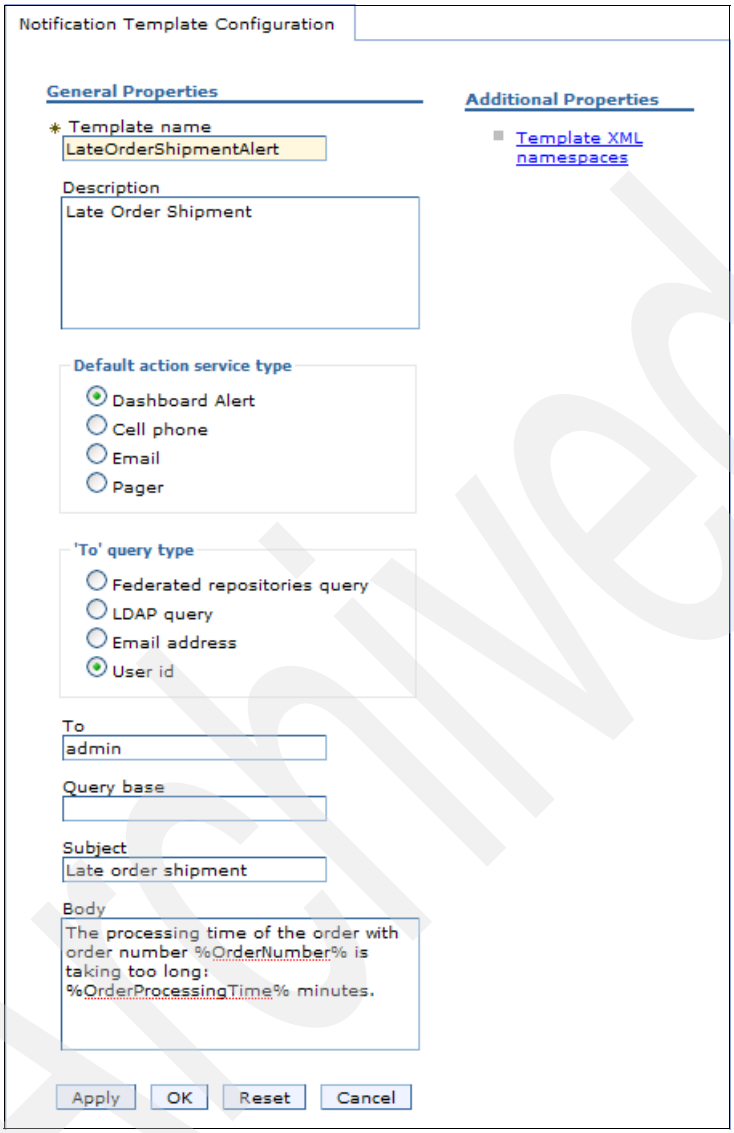


Figure 5-84 Creating an alert template

Archived

3. Add a binding from the situation event to the alert template:
   a. Expand **Applications** → **Monitor Action Services** and click **Installed Situation Event Bindings** (Figure 5-85).
   b. In the Installed Situation Event Bindings panel, click **New**.

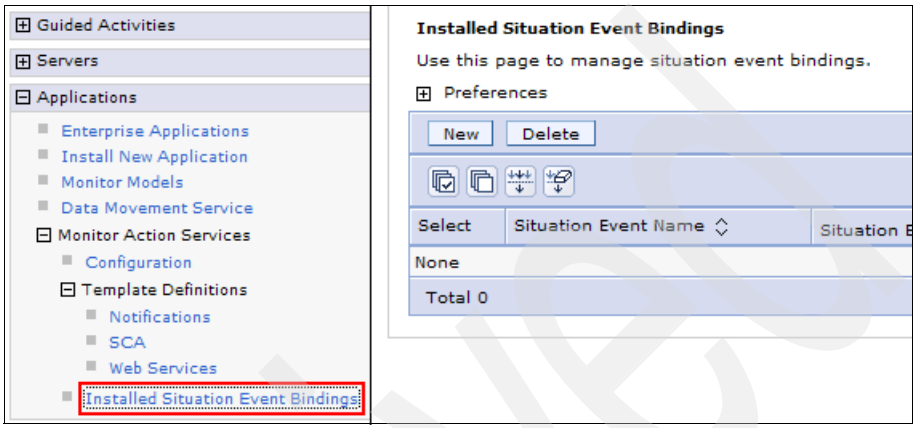


Figure 5-85 Navigating to Installed Situation Event Bindings

4. Under New Situation Event Binding (Figure 5-86), for Situation event name, type `Order handling duration is too long`.

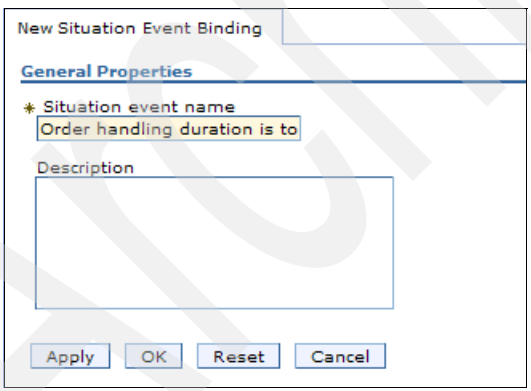


Figure 5-86 Creating a new situation event binding

Archive

The Situation event name must match the value in the BusinessSituationName field of the Late Order Shipped outbound event (Figure 5-87 on page 229). Click **Apply**.

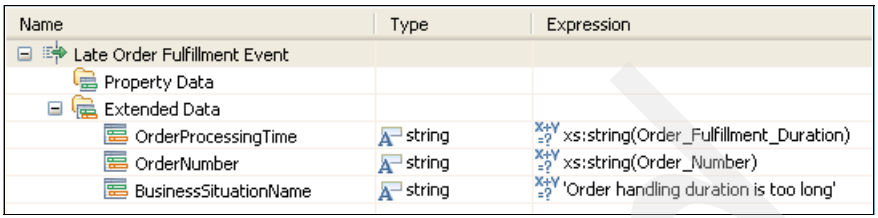


Figure 5-87 Business Situation Name setting

5. Click **Add**.
6. Under Add template to situation event binding (Figure 5-88), for Binding name, type `LateOrderShipmentBinding`, and for Template name, select **LateOrderShipmentAlert**. Click **OK**.

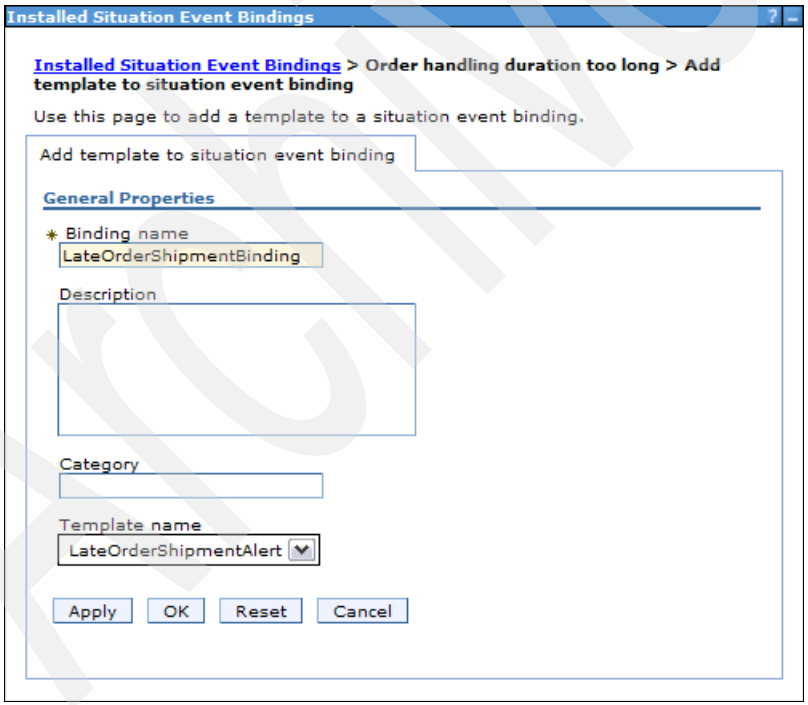


Figure 5-88 Add template to situation event binding page

7. Restart WebSphere Business Monitor.

Archive

Starting process instances

We execute the process instances by using the Business Process Choreographer Explorer. This results in events being sent from WebSphere Process Server to WebSphere Business Monitor.

1. In the Servers view of WebSphere Integration Developer (Figure 5-89), right-click **WebSphere Process Server v.6.1** and select **Launch** → **Business Process Choreographer Explorer**.

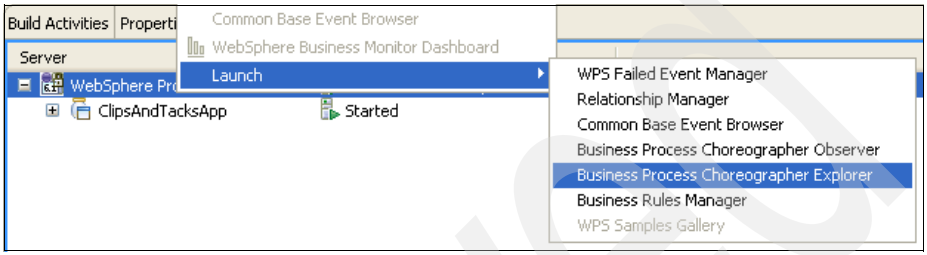


Figure 5-89 Launching Business Process Choreographer Explorer

2. In the Business Process Choreographer Explorer (Figure 5-90), select **Process Templates** → **My Process Templates**.

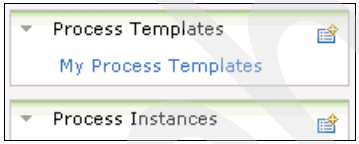


Figure 5-90 Selecting My Process Templates

3. In the My Process Templates panel (Figure 5-91), select the **OrderHandling** check box and click **Start Instance**.

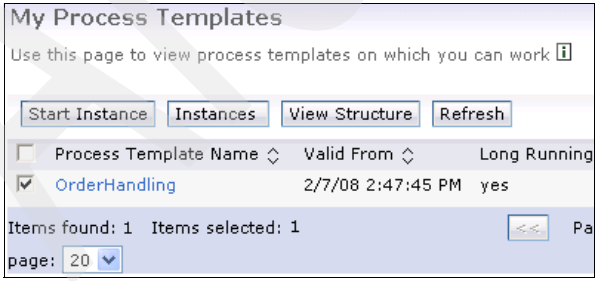


Figure 5-91 Starting an OrderHandling Process Instance

Archive

4. Start the following five process instances by entering the test values as provided for each path. In addition, enter the values for Order number, city, and Country.

   a. Approve without review, available credit > total price, shipped

      i. Classification: `SILVER`
      ii. Available credit: `1000`
      iii. Total price: `800`
      iv. Ship Order task: No required elements

   b. Approve without review, available credit < total price, approve the order, shipped

      i. Classification: `GOLD`
      ii. Available credit: `1000`
      iii. Total price: `1400`
      iv. Review Order task: `APPROVED`
      v. Ship Order task: No required elements

   c. Approve without review, available credit < total price, decline the order, cancelled

      i. Classification: `GOLD`
      ii. Available credit: `1000`
      iii. Total price: `1400`
      iv. Review Order task: `DECLINED`

   d. Do not approve without review, approve the order, shipped

      i. Classification: `SILVER`
      ii. Available credit: `1000`
      iii. Total price: `1400`
      iv. Review Order task: `APPROVED`
      v. Ship Order task: No required elements

   e. Do not approve without review, decline the order, cancelled

      i. Classification: No value required
      ii. Available credit: `200`
      iii. Total price: `800`
      iv. Review Order task: `DECLINED`

   After entering the test values, click the **Submit** button.

Archived

Figure 5-92 shows a sample window before clicking the Submit button.

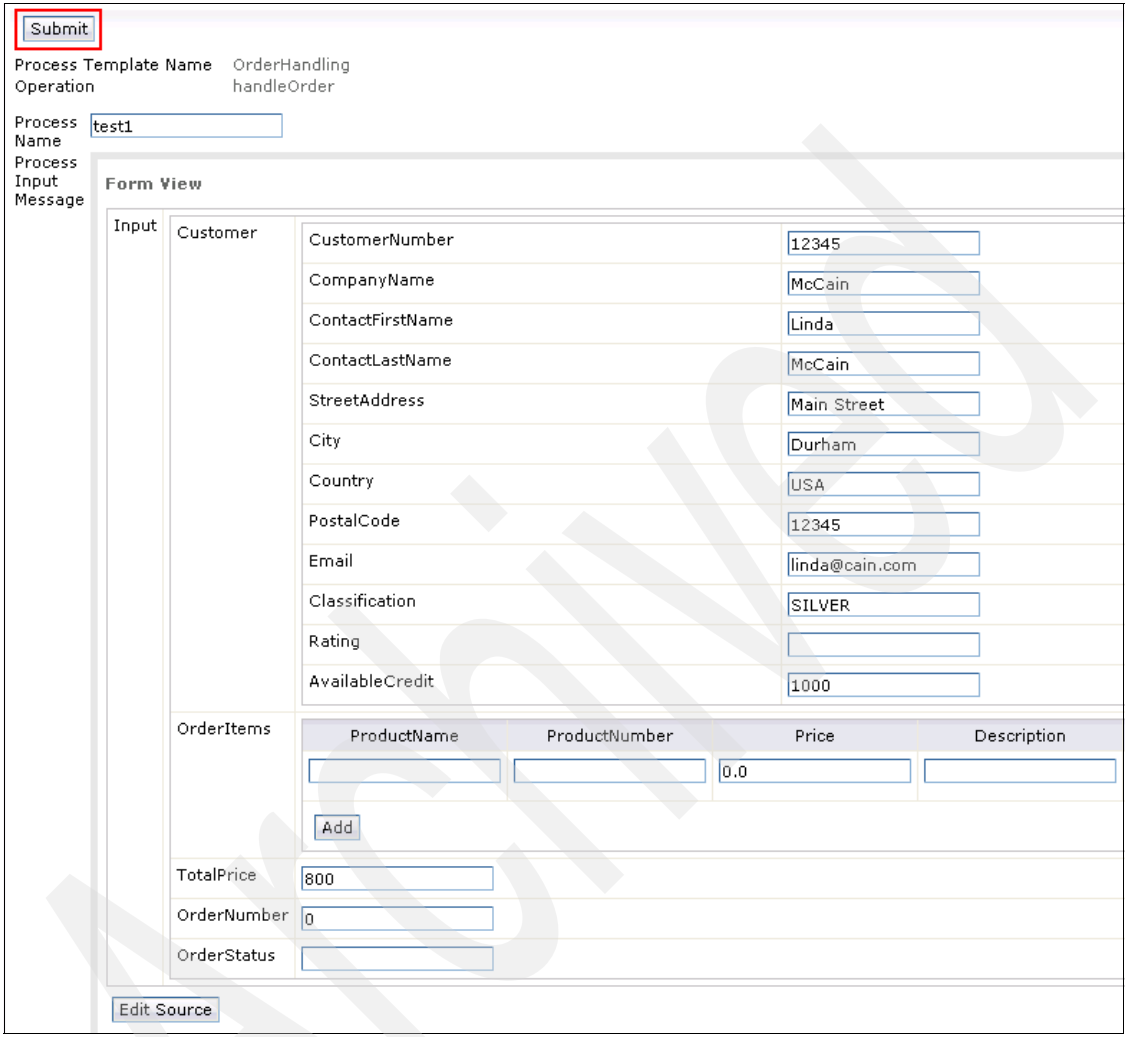


Figure 5-92 Entering process input data

5. To execute the human tasks, select **Task Instances** → **All Tasks** (Figure 5-93).

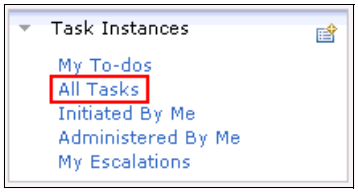


Figure 5-93 Navigating to All Tasks

6. In the All Tasks panel (Figure 5-94), select a task and click **Work on**.

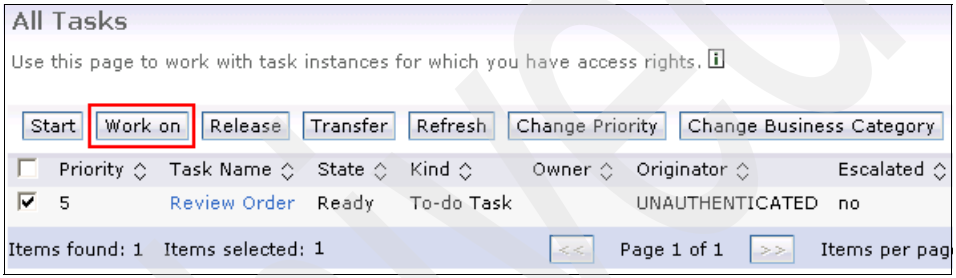


Figure 5-94 Clicking Work on to work on human tasks

Archived

7. Enter an output message as described in the test data set and click the **Complete** button (Figure 5-95).

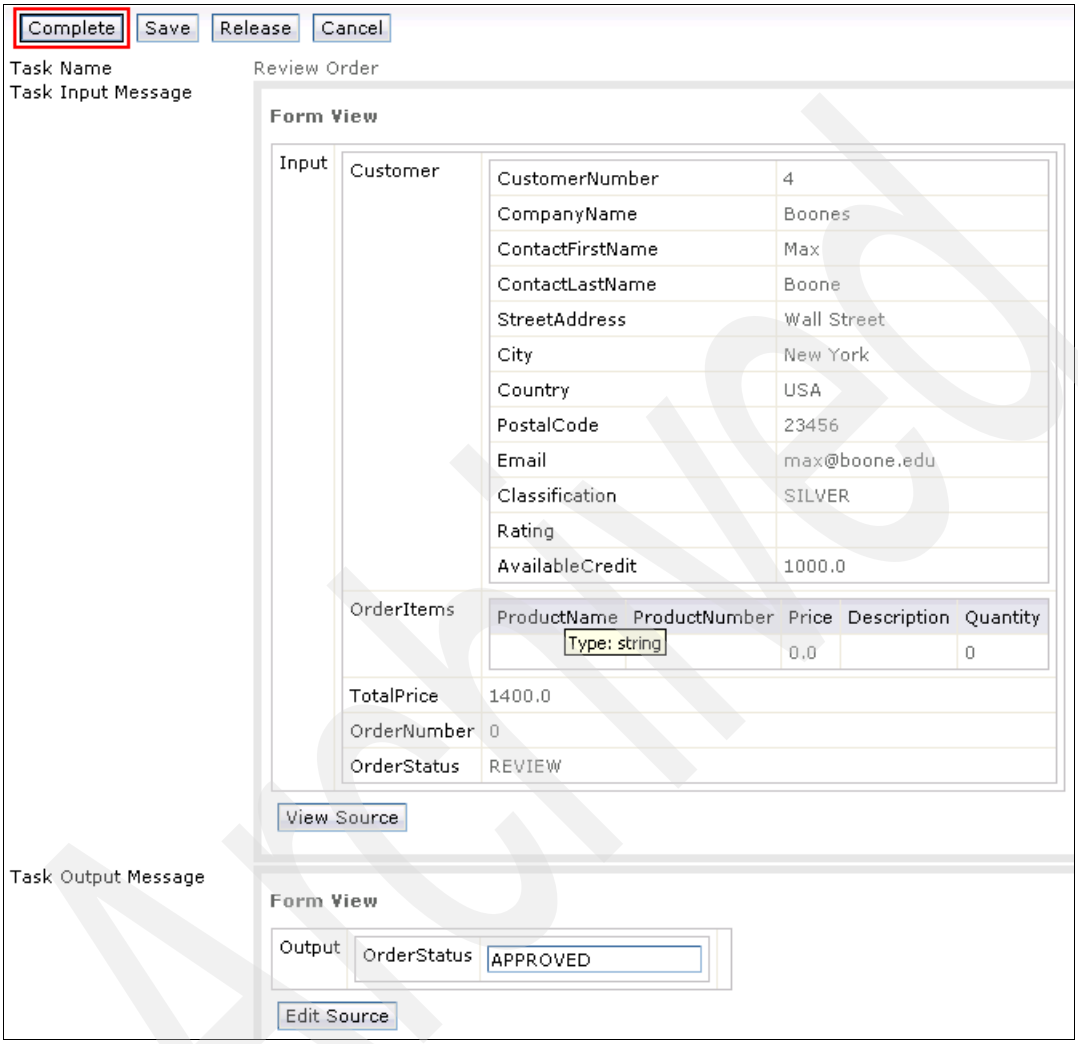


Figure 5-95 Completing a human task

8. Execute at least one process instance with a duration longer than three minutes to test the alert functionality of the WebSphere Business Monitor dashboard.

Archive

Creating the WebSphere Business Monitor Dashboard

In the following sections, we explain the development of a ClipsAndTacks dashboard including the Alphablox views. The five process instances executed in the last section sent events to WebSphere Business Monitor. These events can be displayed in the Business Dashboard of WebSphere Business Monitor.

Creating a dashboard

To create a dashboard:

1. Log in to the Business dashboard.

> **Note:** To access the Business dashboard of WebSphere Business Monitor, enter the following URL:
>
> `http://<host name>:<WC_defaulthost port>/BusinessDashboard`

2. When prompted for the User ID, type `admin` and click **Login** (Figure 5-96). The admin user login is required because all alerts are sent to this user ID.

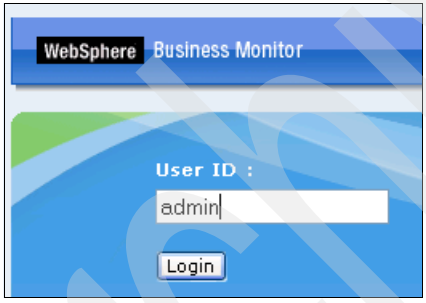


Figure 5-96 Logging into the Business Dashboard

3. Click the **Dashboards** tab and select **New** (Figure 5-97).

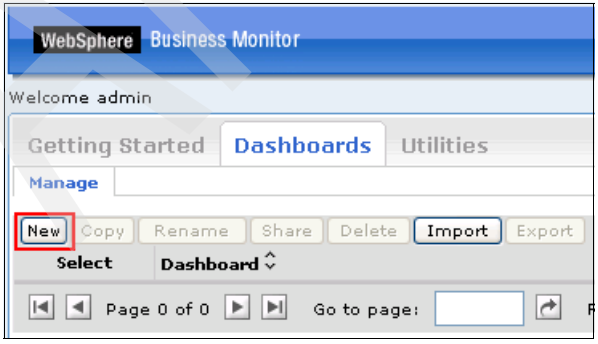


Figure 5-97 Dashboard tab

Archived

4. In the New Dashboard panel (Figure 5-98), for Name, type `ClipsAndTacksDashboard` and click **OK**.

Figure 5-98 Creating a new dashboard

An empty dashboard is displayed in the Business Dashboard view.

Creating an Instances view

To add an Instances view:

1. Click the **Add to Dashboard** link (Figure 5-99) at the top of the Dashboard page.

Figure 5-99 Clicking the Add to Dashboard link

2. In the Add to Dashboard panel (Figure 5-100), select **Instances** and click **OK**.

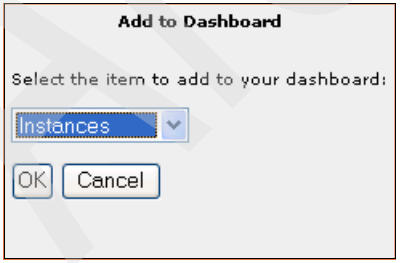


Figure 5-100 Adding the Instance view

Archived

3. In the Instances panel (Figure 5-101):

   a. Click the **Personalize** button.

Figure 5-101 Clicking the Personalize button

   b. Click the **Show/Hide** tab (Figure 5-102). Click >> to copy all metrics from the available list to the Selected list.

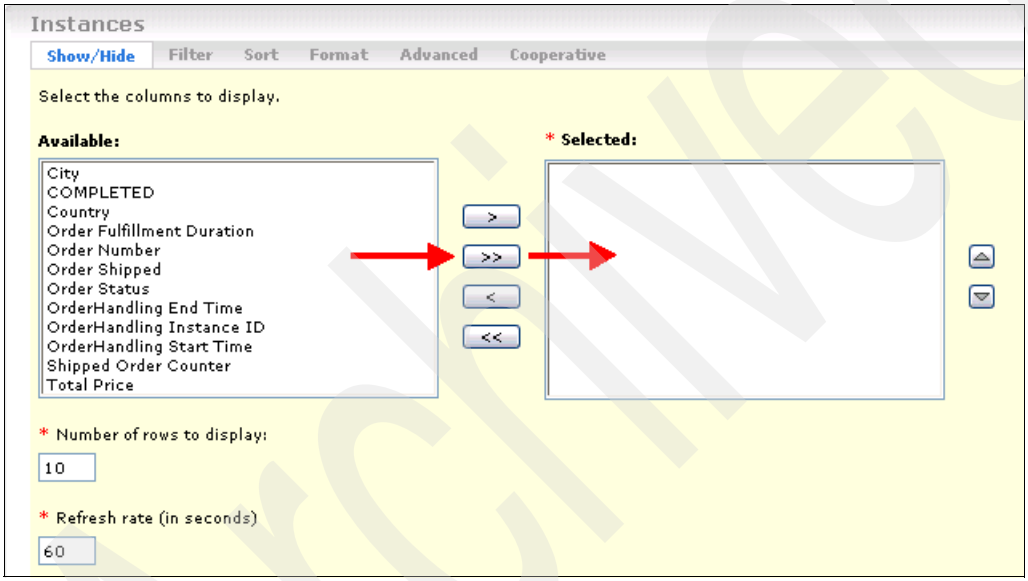


Figure 5-102 Selecting the Instances view items

   c. Click the **Cooperative** tab (Figure 5-103). Select **Cooperative mode** to create a link between this view and other views such as diagrams or alerts.

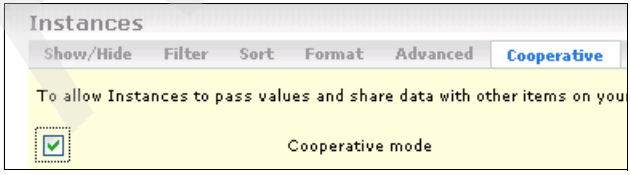


Figure 5-103 Enabling Cooperative mode for Instances view

   d. Click **OK**.

Archived

The Instances view (Figure 5-104) shows a list of monitoring context instances for the events that were emitted by the five test processes.

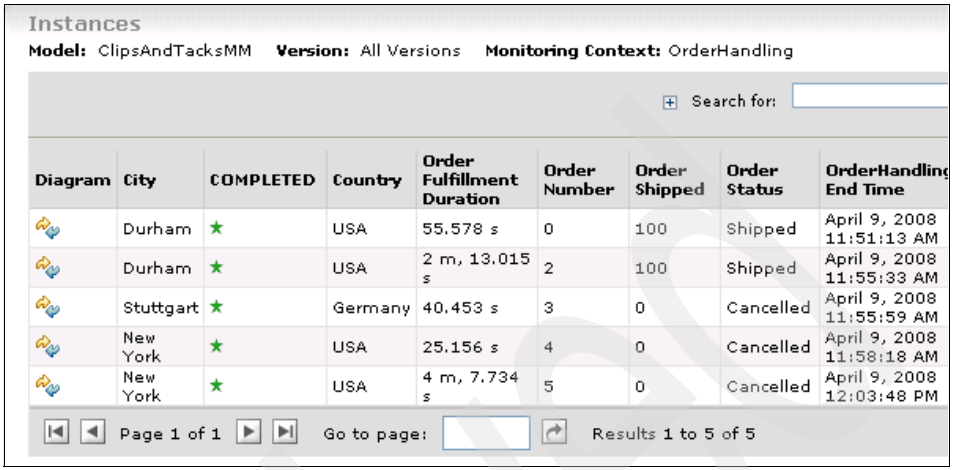


Figure 5-104 Instances view

Creating a Diagrams view for the monitoring context

The Diagrams view for the monitoring context has an enabled cooperative mode. Therefore, it can show the configured SVG diagram for a specific monitoring context instance.

SVG viewer: An SVG viewer must be installed to display the SVG diagrams. If the SVG diagrams do not display correctly, refer to the Adobe® SVG page at the following address:

`http://www.adobe.com/svg/viewer/install/main.html`

1. Click the **Add to Dashboard** link.
2. Select **Diagrams** and click **OK**.

Archive

3. In the Diagrams panel:
   a. Click the **Personalize** button.
   b. Click the **Select a Diagram** tab (Figure 5-105):
      i. For Model, select **ClipsAndTacksMM**.
      ii. Select **Monitoring Context** and select **OrderHandling**.

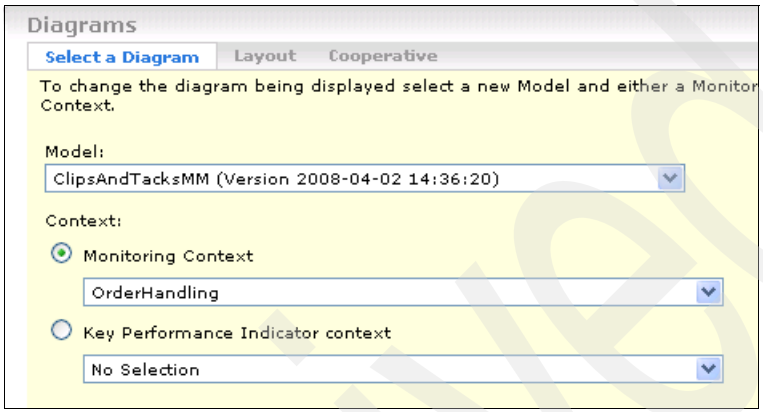


Figure 5-105 Configure Diagrams view

   c. Click the **Cooperative** tab and select **Cooperative mode**.
   d. Click **OK**.

   The generic SVG diagram is displayed in the view without any visual actions.

4. In the Instance view, click the **diagram** icon ( ) in the diagram column to update the monitoring context Diagrams view for this monitoring context.
5. For a shipped order, the Ship Order activity is green (Figure 5-106). For a cancelled order, the Cancel Order And Sent Notification activity is red.

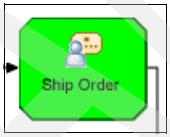


Figure 5-106 Monitoring context Diagrams view color action

6. In the Instance view, click the **Restore Original Content** button to reset the diagram (Figure 5-107).

Figure 5-107 Restore Original Content button

Archived

Creating a KPI view

The KPI view of the dashboard enables a graphical representation of available KPIs. To create this view:

1. Click the **Add to Dashboard** link.
2. Select **KPIs** and click **OK**.
3. In the KPIs panel:
   a. Click the **Personalize** button.
   b. Click the **Select KPIs** tab (Figure 5-108). Expand **ClipsAndTacksMM** to see the available KPI metrics. Select all of them.

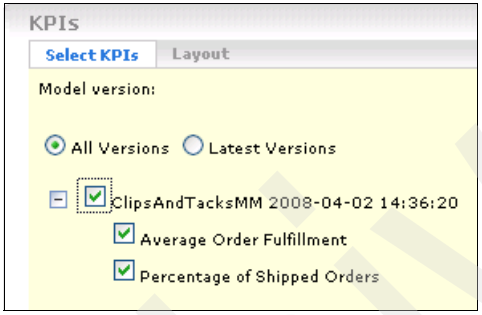


Figure 5-108 Configuring the KPIs view

   c. Switch to the **Layout** tab (Figure 5-109). For Select a display mode, select **Half Gauge**.

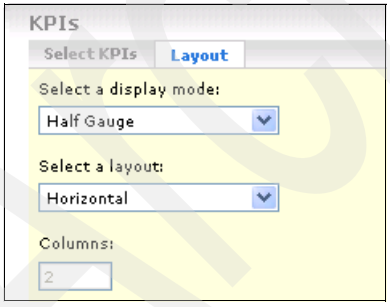


Figure 5-109 Configuring the layout of the KPIs view

   d. Click **OK**.

Archived

The KPIs are displayed as shown in Figure 5-110.

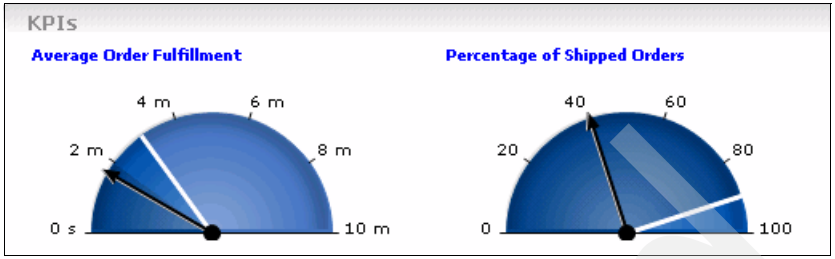


Figure 5-110 KPIs view

Creating a Diagrams view for the KPI context

To create a Diagrams view for the KPI context:

1. Click the **Add to Dashboard** link.
2. Select **Diagrams** and click **OK**.
3. In the Diagrams panel, click the **Select a Diagram** tab (Figure 5-111):
   a. Click the **Personalize** button.
   b. For Model, select the **ClipsAndTacksMM**.
   c. Click **Key Performance Indicator context** and select **ClipsAndTacks KC**.

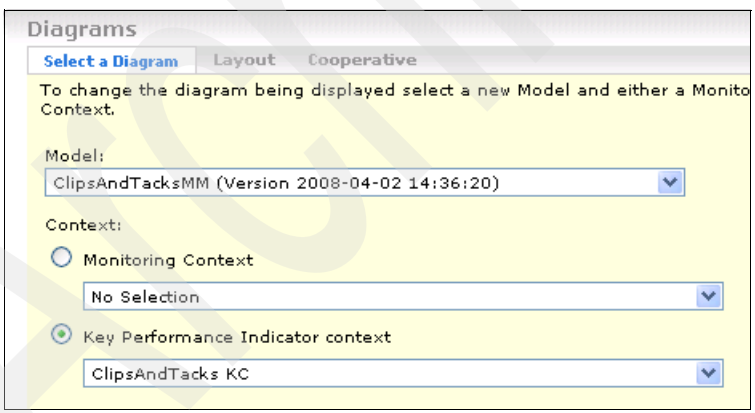


Figure 5-111 Configuring the KPI Diagrams view

   d. Click **OK**.

Archive

The KPI SVG diagram is displayed in the view, indicating the percentage of shipped orders (Figure 5-112).

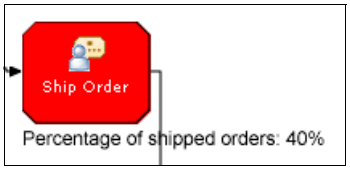


Figure 5-112 KPI Context Diagrams view color action

Creating an Alerts view

To create an Alerts view that displays an alert for the logged in user:

1. Click the **Add to Dashboard** link.
2. Select **Alerts** and click **OK**.

The Alerts are displayed as shown in Figure 5-113.

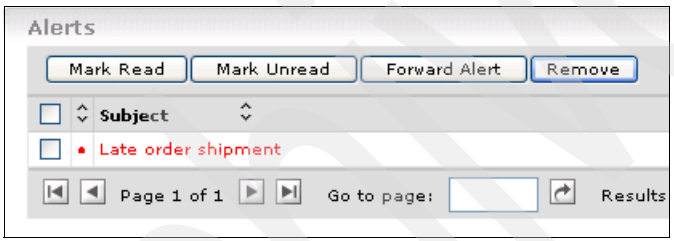


Figure 5-113 Alerts view

Creating a Dimensions view

To create a Dimensions view:

1. Click the **Add to Dashboard** link.
2. Select **Dimensions** and click **OK**.

Archived

3. In the Dimensions panel (Figure 5-114):

   a. Click the **Personalize** button.
   b. For Monitoring Model, select **ClipsAndTacksMM (All Versions)**.
   c. For Monitoring Context, select **OrderHandling**.
   d. Under Available dimensions:

      i. Select **Location** and click **>** to add it to the Row dimensions box.
      ii. Select **Measures** and click **>** add it to the Column dimensions box.

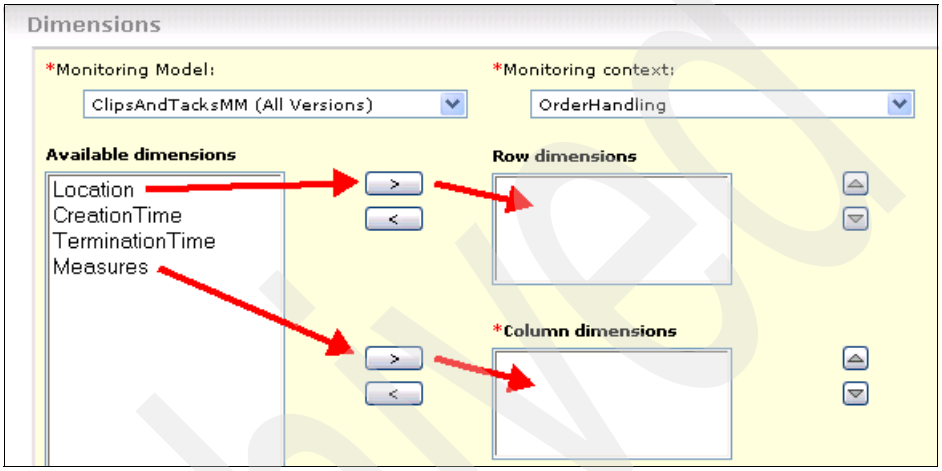


Figure 5-114 Configuring the Dimensions view

   e. Click **OK**.

4. The Dimensions view (Figure 5-115) shows the instances count based on location. Double-click **All Location** to drill down on locations.

Archive

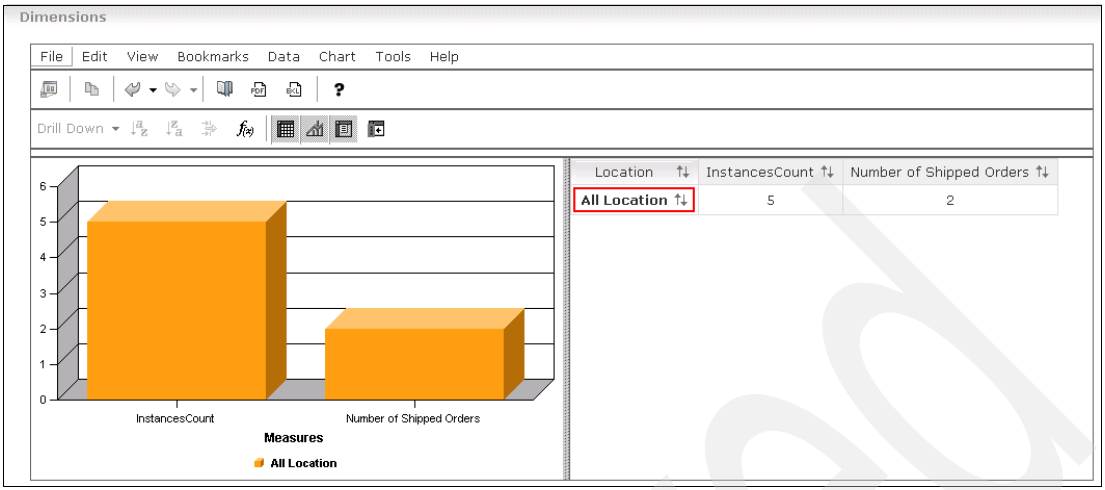


Figure 5-115 Dimensions view

Figure 5-116 shows the drill-down of the locations.

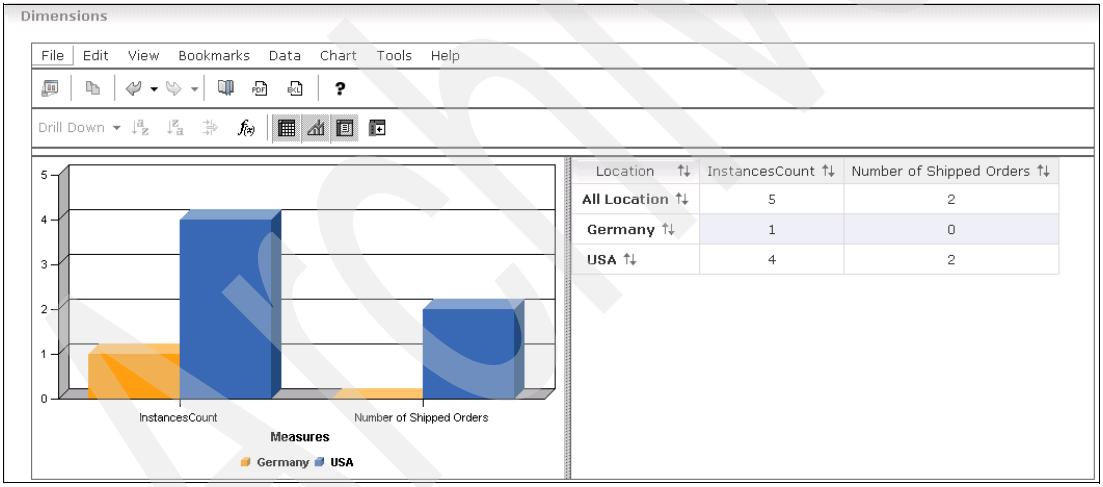


Figure 5-116 Drill-down of a dimension

Creating a Reports view

To create a Reports view:

1. Click the **Add to Dashboard** link.
2. Select **Reports** and click **OK**.

Archived

3. In the Reports panel (Figure 5-117):
   a. Click the **Personalize** button.
   b. For Model, select **ClipsAndTacksMM (All Versions)**.
   c. For Monitoring Context, select **OrderHandling**.
   d. For Dimension, select **Location**.
   e. For Analysis Type, select **Basic**.
   f. For Time range, set From to **Tuesday, January 1, 2008**. Set To to the current date.
   g. For Frequency, select **Monthly**.
   h. For Modeled Measures, click the **InstancesCount and Number Of Shipped Orders** check box.
   i. Click **OK**.

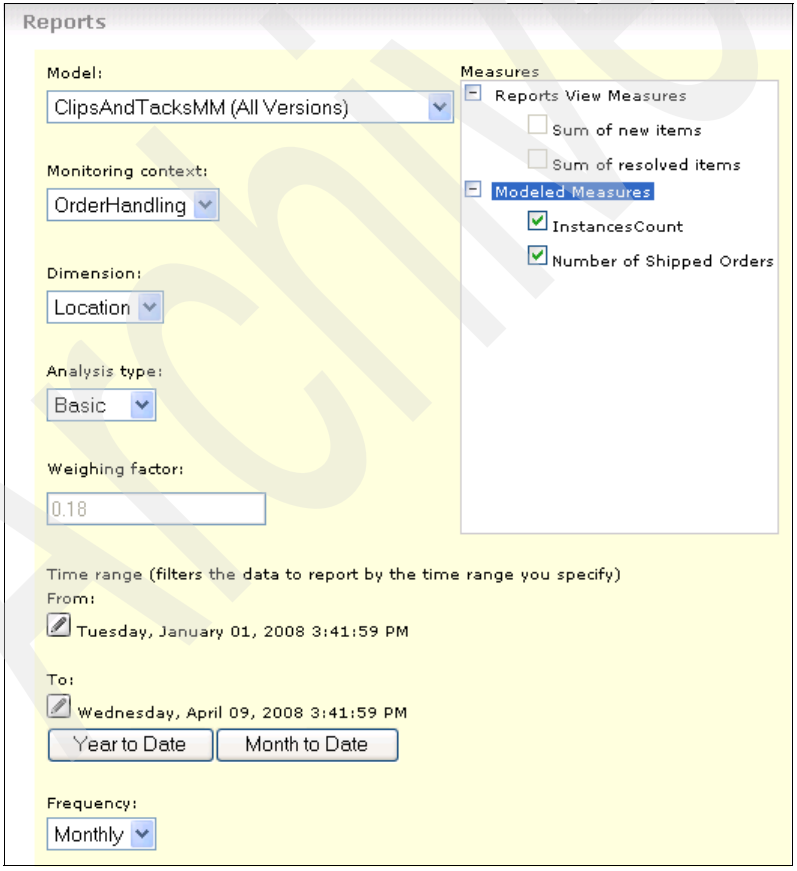


Figure 5-117 Configuring the Reports view

Archived

4. The Reports view shows the instances count based on the location relative to the creation time as shown in Figure 5-118. Double-click **All Location** to drill down by location.

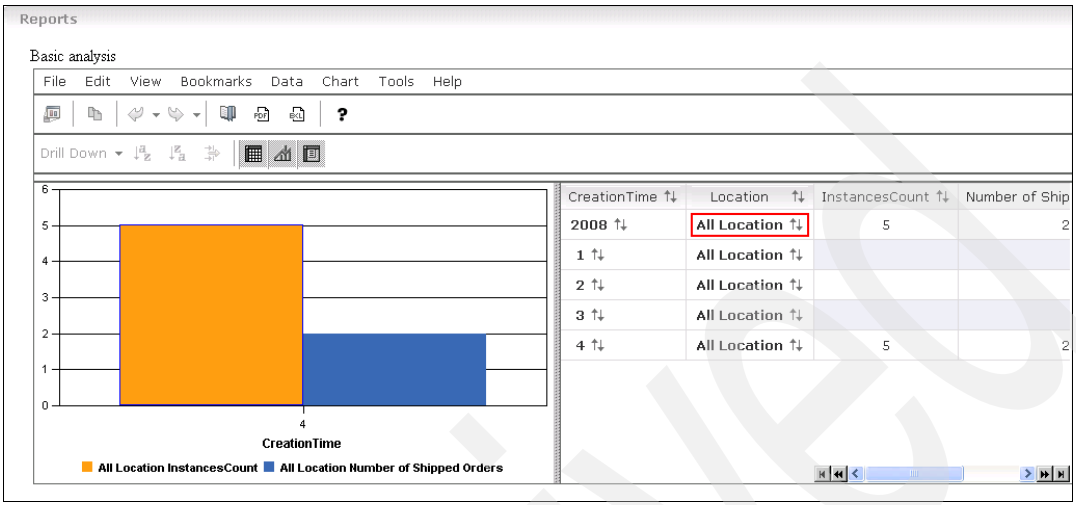


Figure 5-118 Reports view

Figure 5-119 shows the drill down by location.

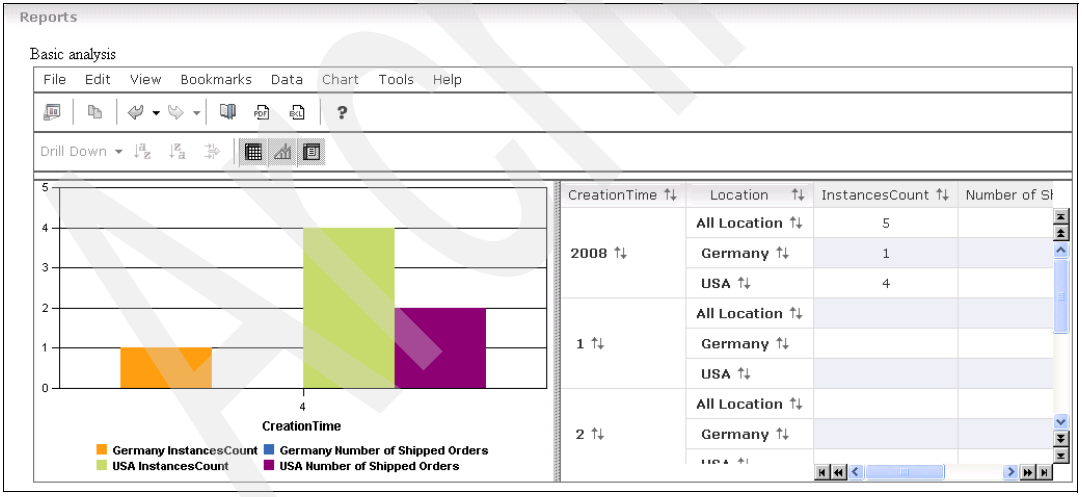


Figure 5-119 Drilling down a dimension in the Reports view

Archive

5.4 Summary

In this chapter, we discussed the integration of WebSphere Process Server into WebSphere Business Monitor. We explained the ClipsAndTacks Order Handling process implementation. We also described all steps required to generate and extend a monitor model based on that process by using the WebSphere Business Monitor Toolkit. Then we deployed the process and the monitor model to the WebSphere Process Server and WebSphere Business Monitor server. Finally, we created a WebSphere Business Monitor dashboard to monitor executed process instances.

Archived

Archived

6

WebSphere ESB and IBM Adapters

Adapters are commonly used together with enterprise service bus (ESB) technologies to provide connectivity between enterprise information systems (EISs). They provide a consistent and independent mechanism for accessing EISs.

In this chapter, we discuss how the IBM Adapters and the WebSphere Enterprise Service Bus can be employed for business activity monitoring (BAM). We also look at the creation of an associated monitor model that describes metrics and key performance indicators (KPI) that are of interest to ClipsAndTacks Office Supplies Pte. Ltd.

© Copyright IBM Corp. 2008. All rights reserved.

6.1 Business activity monitoring with IBM Adapters

In this section, we discuss the following topics:

- 6.1.1, “IBM Adapters”
- 6.1.2, “WebSphere ESB” on page 251
- 6.1.3, “IBM Adapters and WebSphere ESB” on page 252
- 6.1.4, “Service component monitoring” on page 252
- 6.1.5, “Event generation options” on page 254

6.1.1 IBM Adapters

Adapters enable inbound and outbound connectivity between the EIS and Service Component Architecture (SCA)-based applications hosted by WebSphere Enterprise Service Bus or WebSphere Process Server. The IBM Adapters that we describe include the J2EE Connector Architecture (JCA) v1.5-compliant WebSphere Adapter and the WebSphere Business Integration Adapters, which are based on the WebSphere Business Integration Adapter framework. The WebSphere Adapters are deployed as part of a J2EE application as an embedded resource adapter (Figure 6-1).

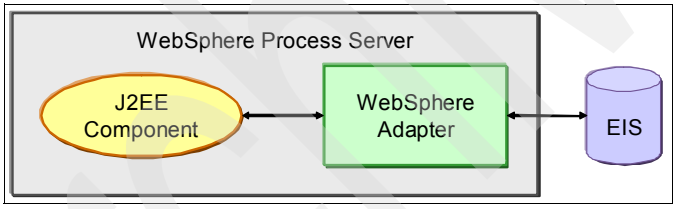


Figure 6-1 WebSphere Adapter

The WebSphere Business Integration Adapters reside outside of the application layer and communicate with the server by using the Java Message Service (JMS) transport layer as shown in Figure 6-2.

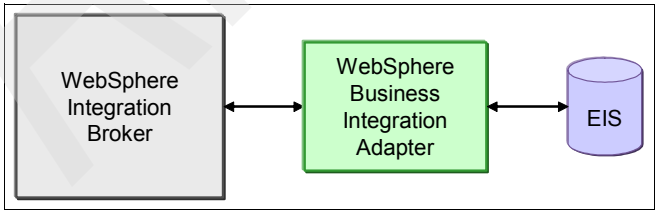


Figure 6-2 WebSphere Business Integration Adapter

In the following sections, we discuss the use of WebSphere Adapters.

6.1.2 WebSphere ESB

WebSphere Enterprise Service Bus is designed to provide an ESB for IT environments built around open standards and service-oriented architecture (SOA). It delivers easy-to-use functionality that is built on the messaging and Web services technologies of WebSphere Application Server.

The development tool for WebSphere Enterprise Service Bus is WebSphere Integration Developer.

Architecture

WebSphere Enterprise Service Bus provides uniform invocation and data-representation programming models, as well as, monitoring capabilities for the following components that run on WebSphere Enterprise Service Bus:

- Service Component Architecture

  On top of the infrastructure provided by WebSphere Application Server, WebSphere Enterprise Service Bus implements an SCA-based mediation framework. SCA is designed around components that encapsulate services that can be invoked. This mediation framework provides for enhanced flexibility, encapsulation, and reuse.

- Service Message Object (SMO)

  Service Data Objects (SDOs) and business objects define the data that flows between defined components in SCA. SMOs are enhanced SDOs and provide an abstraction layer for processing and manipulating messages exchanged between services.

  Mediation components are typically concerned with the flow of messages through the infrastructure and not only with the business content of the message. The information that governs their behavior is often held in headers that flow with the business message. Therefore, the SMO pattern for SDO is introduced to support this pattern.

- Common Event Infrastructure (CEI)

  WebSphere Enterprise Service Bus uses the CEI to provide event management services. The format of those events is defined by the Common Base Event specification.

Archived

6.1.3 IBM Adapters and WebSphere ESB

The IBM Adapters are used to receive the business events from the EIS and represent it as a common business object. The WebSphere Enterprise Service Bus provides the mediation capability through the mediation module to augment or filter events that are of interest to the organization.

WebSphere Enterprise Service Bus also provides the infrastructure to emit Common Base Events as required by WebSphere Business Monitor. While we discuss WebSphere ESB in this chapter, this same concept applies to WebSphere Process Server.

6.1.4 Service component monitoring

WebSphere Enterprise Service Bus operates on top of an installation of WebSphere Application Server. Consequently, it uses much of the functionality of the application server infrastructure for monitoring system performance and troubleshooting.

However, WebSphere Enterprise Service Bus goes beyond the traditional function of monitoring to aid in system administration functions, such as performance tuning and problem determination. For information about the system monitoring functionality of WebSphere Enterprise Service Bus, refer to the monitoring and troubleshooting topics found in the WebSphere Application Server, Version 6.1 Information Center at the following address:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp`

WebSphere Enterprise Service Bus can monitor the processing of service components within the applications deployed on the system. This capability provides a view of the overall processing flow of the interconnected components and ensures that the system is producing the expected result (Figure 6-3 on page 253).

Archive

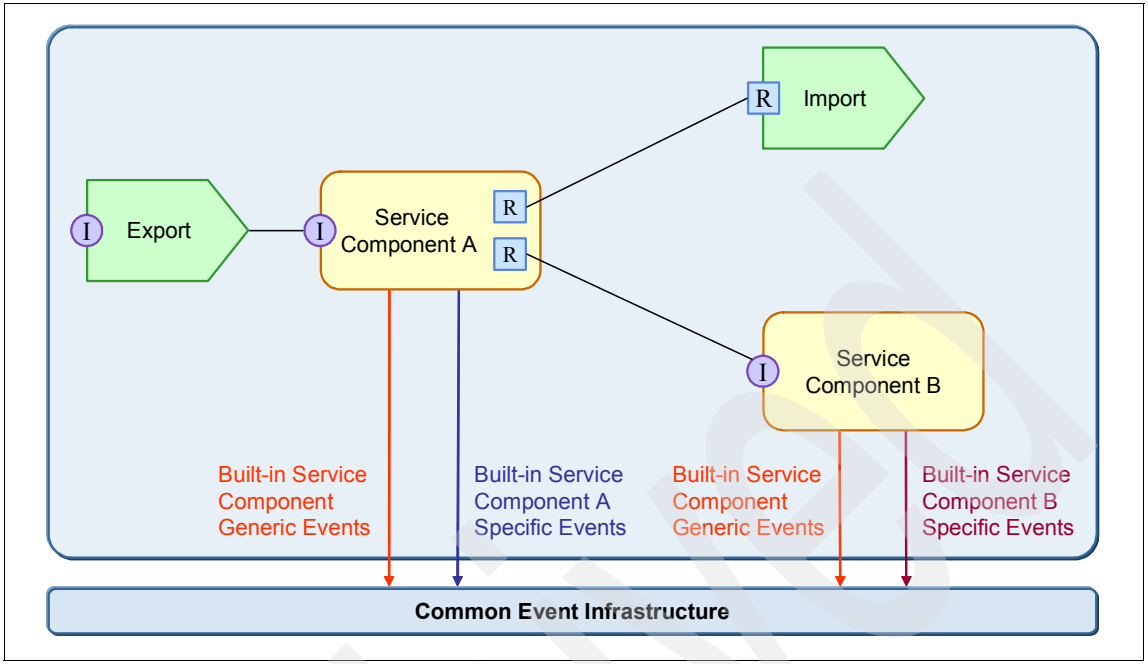


Figure 6-3 Service component monitoring

How service component monitoring works

Service component monitoring entails monitoring a certain point that is reached during the processing of the service components. These points are referred to as an *event point*. Event points encapsulate information about the specific functions of a service component type and the nature of the event.

Event nature

Event nature refers to the situations that are required to generate events during the processing of service components. Common service component event natures are ENTRY, EXIT, and FAILURE event natures. However, there are many other component and element specific event natures.

As an example of how events are defined for a specific component kind, the MAP service component can fire common event natures of ENTRY, EXIT, and FAILURE.

Archived

However, it also includes a *Transformation element kind*, which defines a specific type of functionality within the MAP component kind. The Transformation element kind fires events with ENTRY, EXIT, and FAILURE natures. Consequently, the MAP service component kind can fire up to six different events depending on the combination of elements and natures that you specify (Table 6-1).

Table 6-1 Map events

Event name	Event nature	Event content	Type
map:WBI.MAP.ENTRY	ENTRY	N/A	N/A
map:WBI.MAP.EXIT	EXIT	N/A	N/A
map:WBI.MAP.FAILURE	FAILURE	FailureReason	Exception
map:WBI.MAP.Transformation.ENTRY	ENTRY	N/A	N/A
map:WBI.MAP.Transformation.EXIT	EXIT	N/A	N/A
map:WBI.MAP.Transformation.FAILURE	FAILURE	FailureReason	Exception

Event catalog

The list of all service components, their elements, and their event natures are in the event catalog. For details about the event catalog, refer to the WebSphere Enterprise Service Bus v6.1 Information Center at the following Web address:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wesb610.doc/doc/cmon_event_catalog.html

6.1.5 Event generation options

The following event generation options are available in WebSphere Enterprise Service Bus:

- Built-in component-generic events

  Components can generate built-in events that are common to all components when invoked.

- Built-in component-specific events

  Components can generate built-in events that are specific to the behavior of a component when invoked.

- Mediation flow custom events

  Mediation flows can generate custom events by using the Event Emitter primitive.

- Programmatic custom events

  Components can generate custom events by using the Common Base Event and CEI APIs.

In this book, we discuss Built-in component-generic events and Mediation flow custom events in detail. For further discussion about this topic, refer to the developerWorks article, “Event Correlation Options for Business Activity Monitoring with WebSphere Business Monitor v6.0.2,” at the following address:

http://www.ibm.com/developerworks/websphere/library/techarticles/0709_erpenbach/0709_erpenbach.html

6.2 Scenario description

In the first part of the chapter, we show how the IBM Adapters can be used to retrieve events from an EIS and the monitoring of built-in service component events in WebSphere Enterprise Service Bus. In the second part of the chapter, we look at the generation of custom events by using the Event Emitter primitive in the mediation flow component.

In the ClipsAndTacks scenario, we have a common Order database that is accessed by multiple systems across the organization. For example, the Order database is accessed by both WebSphere Process Server and WebSphere MQ Workflow. This database contains information about the status of customers’ orders. Entries in the database tables are created when orders are placed and updated subsequently when the orders are shipped or cancelled (Figure 6-4).

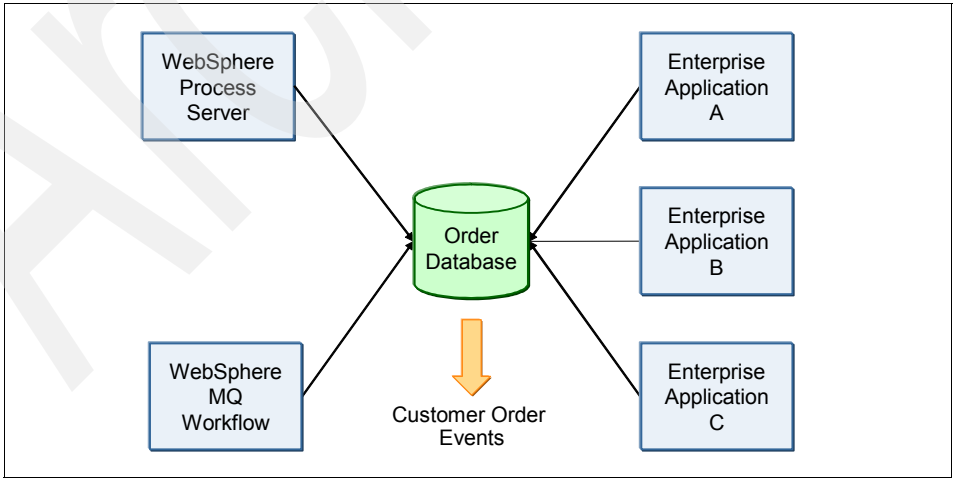


Figure 6-4 Order database

ClipsAndTacks wants to measure the number and types of orders that have arrived based against a set of KPIs. These KPIs include the total number of orders received and the number of large orders.

Event monitoring of the database: The Order database is accessed by both WebSphere Process Server and WebSphere MQ Workflow. From a design perspective, we could choose to monitor the events that are emitted as part of the process models whenever WebSphere Process Server or WebSphere MQ Workflow updates the database. However, we take the approach of monitoring the database itself for events. We discuss this in detail in 6.7, “Design considerations” on page 347. This is another example where the event sources are managed independently from the process engine.

6.3 Preparing the environment for the sample scenario

In this section, we explain how to set up the development environment for the sample scenario and the EIS.

6.3.1 Setting up the development environment

Before you proceed, ensure that the WebSphere Business Monitor Toolkit is installed on WebSphere Integration Developer and you have a monitor profile running on either WebSphere Process Server or WebSphere Enterprise Service Bus. For instructions on how to configure this, refer to Chapter 3, “Using WebSphere Business Monitor” on page 55.

This environment was created by using the following applications:

- WebSphere Integration Developer v6.1.0.003
- WebSphere Business Monitor Toolkit v6.1.1
- DB2 Enterprise Server Edition Version V9.1

6.3.2 Setting up the EIS

In our scenario, we use a DB2 Universal Database as the Order EIS system. In this section, we create the database and the associated event and order tables.

Archive

Creating the Order database

To create the Order database:

1. Ensure that DB2 is started. At a command prompt, type the following command:

   ```
   db2cmd
   ```

2. In the DB2 Command Line Processor, type the following commands:

   ```
   db2 create database ORDER
   ```

Creating the event and order tables

To create the event and order tables:

1. Run the OrderDB.sql script (Example 6-1), which is in the /AdapterWESB/OrderDB.sql directory of the additional materials. For information about how to download the additional materials, see Appendix B, “Additional material” on page 801.

Example 6-1 Listing of OrderDB.sql

```
CREATE TABLE ORDER
(
      onumber VARCHAR (10) NOT NULL PRIMARY KEY,
      fname VARCHAR (20),
      lname VARCHAR (20),
      tctry VARCHAR (20),
      tprice VARCHAR (20),
      ostatus VARCHAR (10)
);

CREATE TABLE WBIA_JDBC_EventStore
(
      event_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY (START
      WITH 1, INCREMENT BY 1, NO CACHE) PRIMARY KEY,
      xid VARCHAR(200),
      object_key VARCHAR(80) NOT NULL,
      object_name VARCHAR(40) NOT NULL,
      object_function VARCHAR(40) NOT NULL,
      event_priority INTEGER NOT NULL,
      event_time TIMESTAMP default CURRENT TIMESTAM NOT NULL,
      event_status INTEGER NOT NULL,
      event_comment VARCHAR(100)
);

CREATE TRIGGER event_create
AFTER INSERT ON ORDER REFERENCING NEW AS N
```

Archived

```
FOR EACH ROW MODE DB2SQL
INSERT INTO wbia_jdbc_eventstore (object_key, object_name,
object_function, event_priority, event_status)
VALUES (N.onumber, 'Db2adminOrderBG', 'Create', 1, 0);
```

2. Navigate to the directory, where you saved the file and run the following commands:

```
db2 connect to Order user db2admin using db2admin
db2 -tvf OrderDB.sql
```

In our example, the user name is *db2admin* and the password is *db2admin*.

6.4 Scenario 1: Monitoring the built-in service component events

In this scenario, we have the following goals for the fictitious company ClipsAndTacks:

- Employ the built-in service component events in WebSphere Enterprise Service Bus for BAM.
- Use the WebSphere Adapters and WebSphere Enterprise Service Bus to emit events from nonmanaged process choreographed event sources.
- Set up the WebSphere Adapters and WebSphere Enterprise Service Bus for BAM.

Archived

Figure 6-5 illustrates the sequence of activities in this scenario. First, the records are created in the Order database. Next the WebSphere Adapter detects the occurrence of the newly created records. Finally, a Java SCA component deployed on WebSphere Enterprise Service Bus emits the events as Common Base Events.

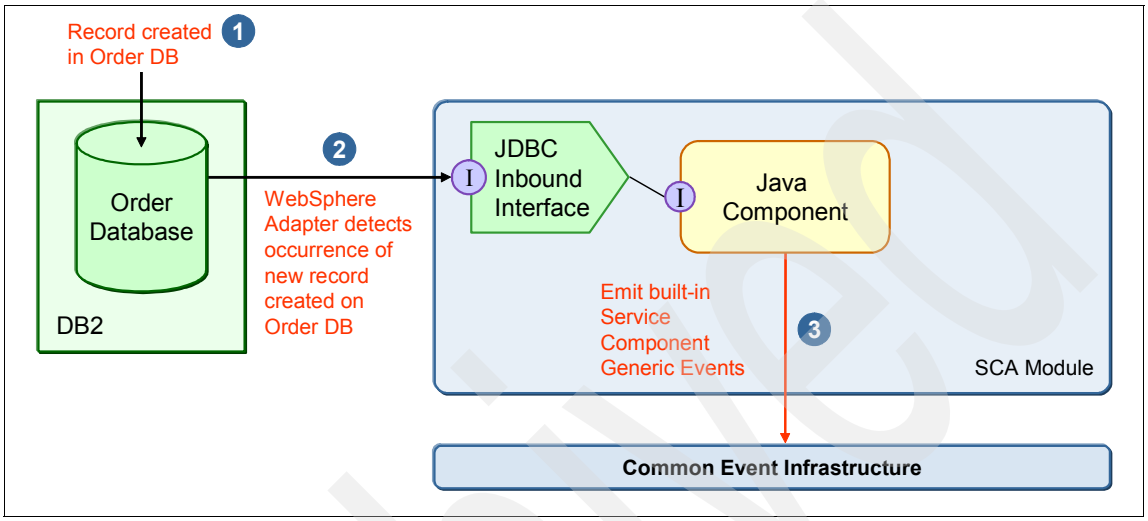


Figure 6-5 Flow of scenario 1: Monitoring the built-in service component events

Archived

6.4.1 Creating a module that contains the WebSphere Adapter

To create the module that contains the WebSphere Adapter:

1. Start WebSphere Integration Developer and ensure that you are currently in the Business Integration perspective (Figure 6-6). If you are not, follow these steps:
   a. In the title bar, select **Window** → **Open Perspective** → **Other**.
   b. Select **Business Integration** (default) and click **OK**.

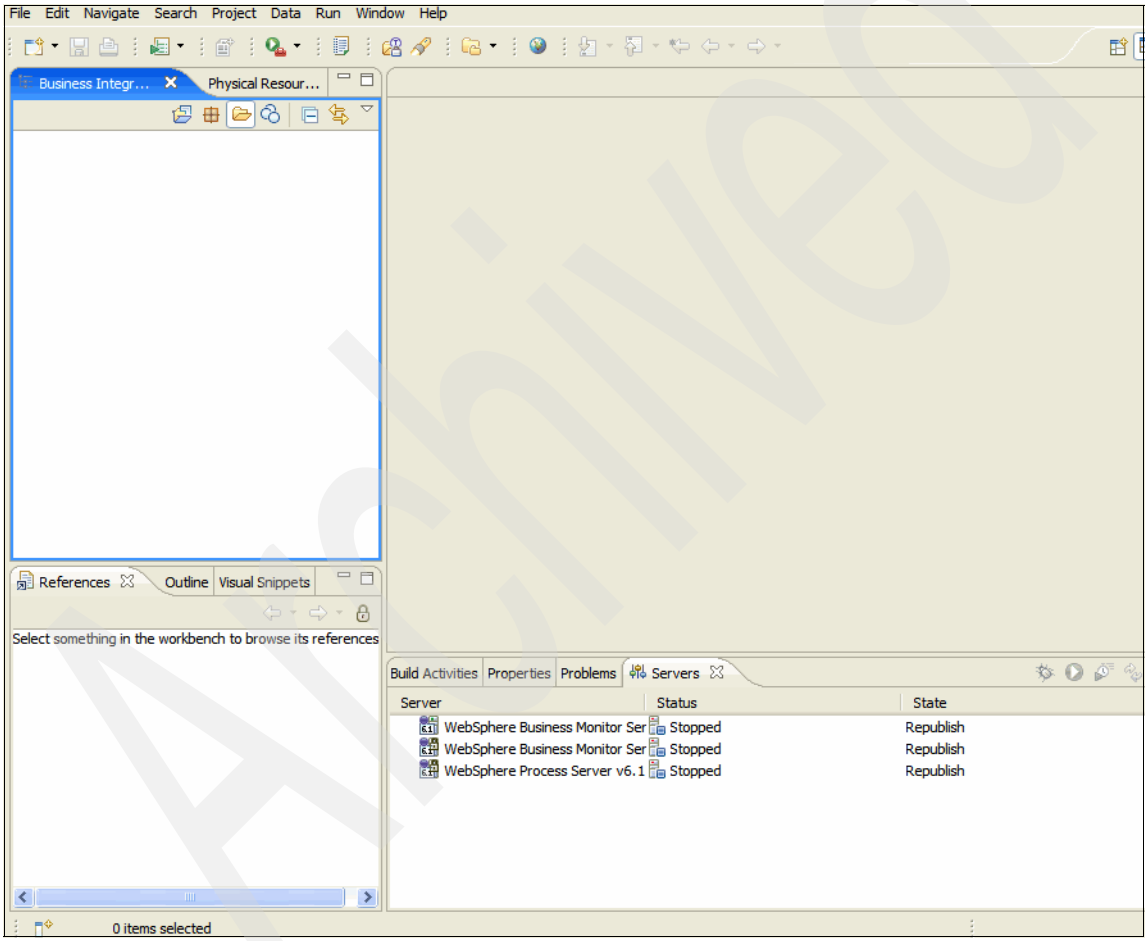


Figure 6-6 Business Integration perspective

Archive

2. In the Business Integration view (Figure 6-7), right-click and select **New** → **Mediation Module**.

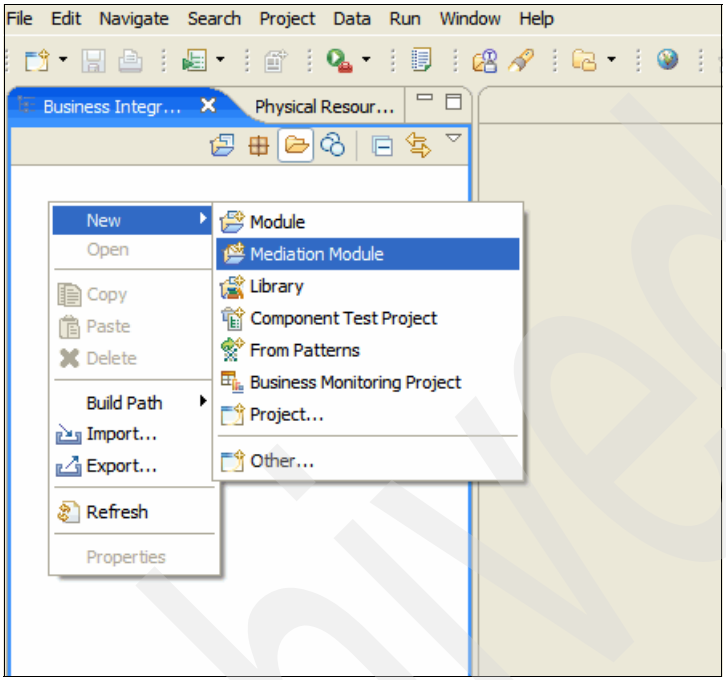


Figure 6-7 Creating a new mediation module

Archived

3. In the New Mediation Module window (Figure 6-8):
   a. For Module name, type `ClipsAndTacks`.
   b. Clear the **Create mediation component** option.
   c. For Target runtime, select **WebSphere ESB Server v6.1**.
   d. Click **Finish** to create the new module.

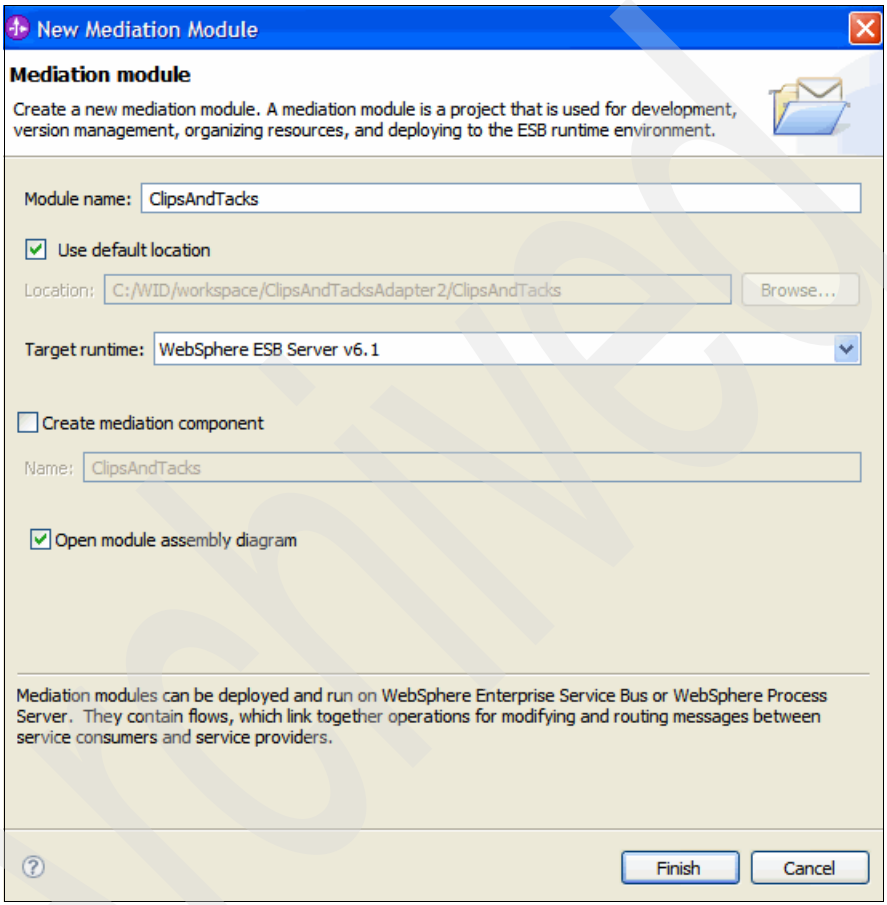


Figure 6-8 Creating the ClipsAndTacks module

Configuring the WebSphere Adapter for JDBC

Run the external service wizard to specify business objects, services, and configuration for the scenario:

1. In the Business Integration view, right-click the newly created **ClipsAndTacks** module and select **New** → **External Service**.
2. In the External Service window, select **Adapters** and click **Next**.

3. In the next window, select **IBM WebSphere Adapter for JDBC** and click **Next**.
4. In the External Service – Adapter Import window (Figure 6-9), for Target runtime, select **IBM WebSphere ESB Server v6.1** and click **Next**.

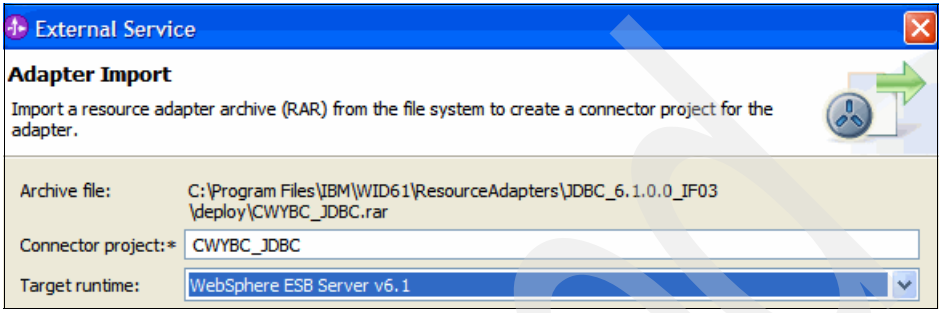


Figure 6-9 Adapter Import

Archived

5. In the External Service – Required Files and Libraries window (Figure 6-10):
   a. Click **Add** to add the JDBC driver JAR files. The JAR files are in <DB2 INSTALL ROOT>/java. In our sample, the JAR files are in C:\Program Files\IBM\SQLLIB\java.
   b. Select the **db2jcc** and **db2jcc_licence_cu.jar** files.
   c. Click **Next**.

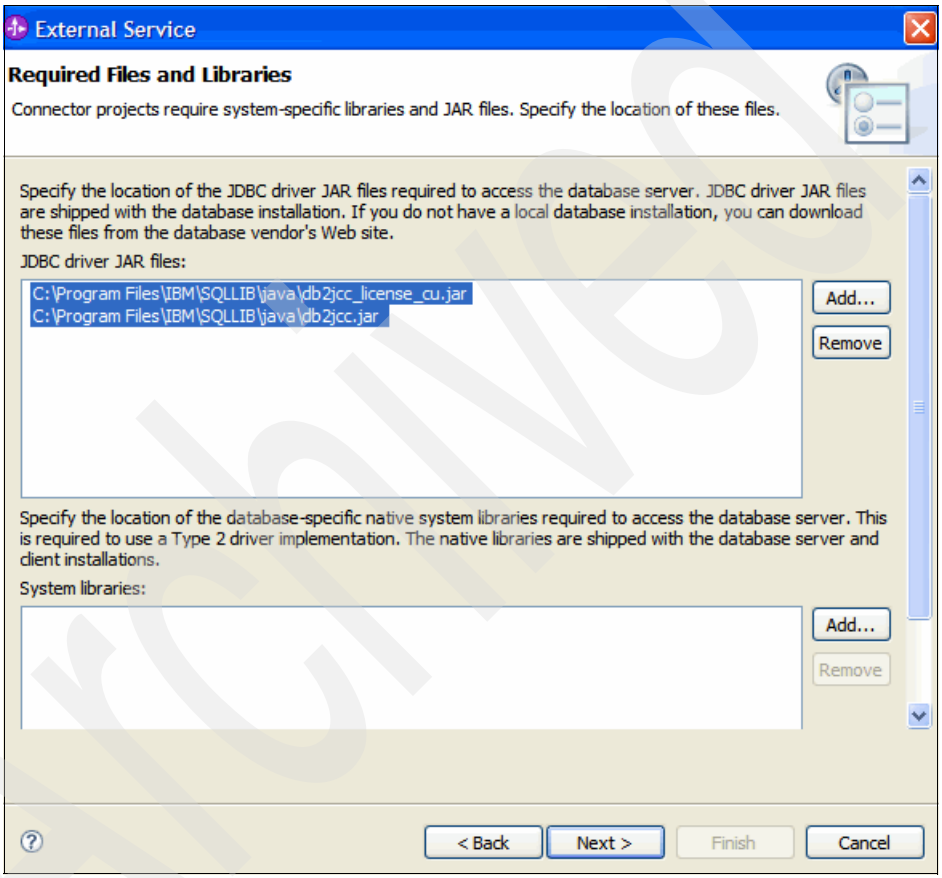


Figure 6-10 Selecting the JDBC driver JAR files

Archived

6. In the External Service – Processing Direction window (Figure 6-11), based on the type of processing the adapter is required to perform, select either Inbound or Outbound. In our example, select **Inbound** because we are passing data from the WebSphere Adapter to the Java Component. Click **Next**.

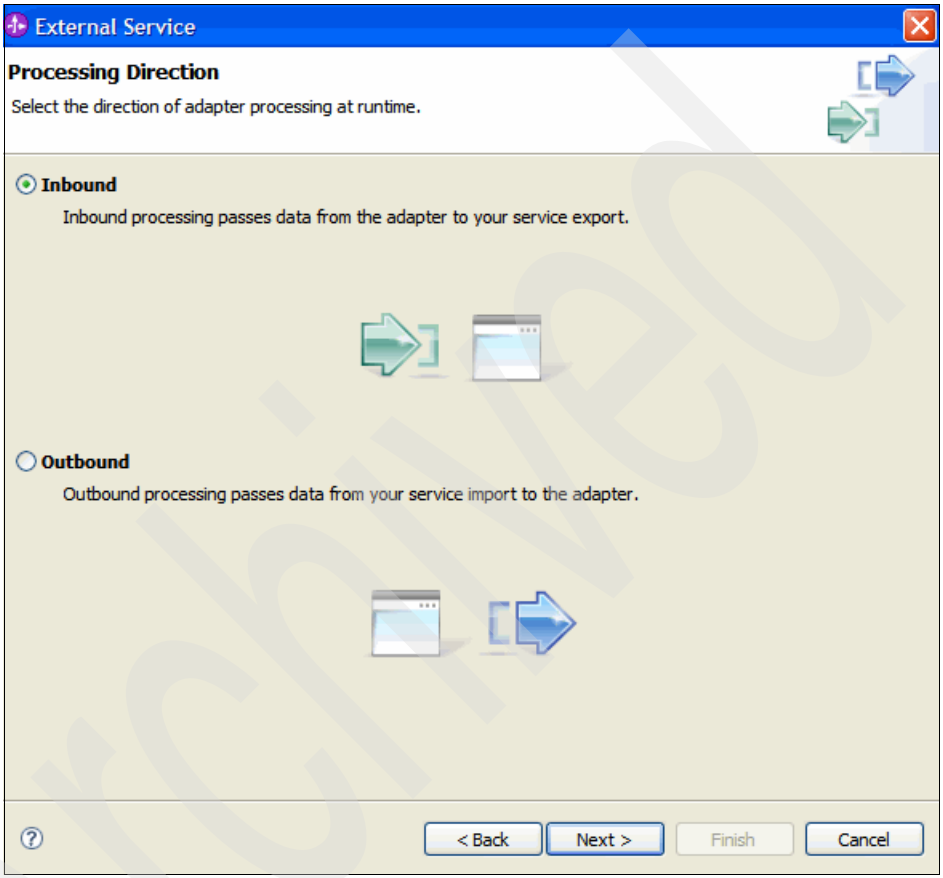


Figure 6-11 Selecting Inbound processing

Archived

7. In the External Service – Discovery Configuration window (Figure 6-12), from the database list, select **DB2 UDB**. Then complete the properties by using the values from Table 6-2. The user name and password might differ for your environment. Click **Next**.

Table 6-2 DB2 Connection properties

Properties	Value
Database	ORDER
Host name	localhost
User name	db2admin
Password	db2admin

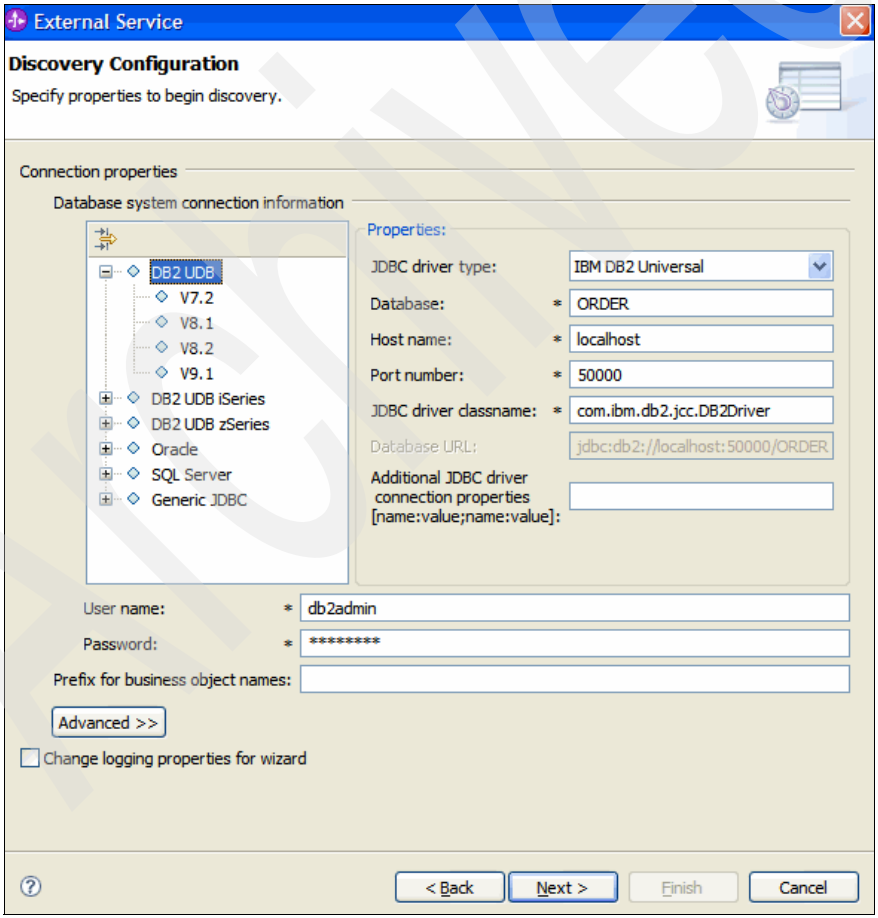


Figure 6-12 Connection properties

8. In the External Service – Object Discovery and Selection window (Figure 6-13), discover the available business objects and services on the database. Click **Run Query** to find the table list. Under Discovered objects, select **DB2ADMIN** → **Tables** → **ORDER** and click the **>** button to add ORDER to the objects to be imported. Click **Next**.

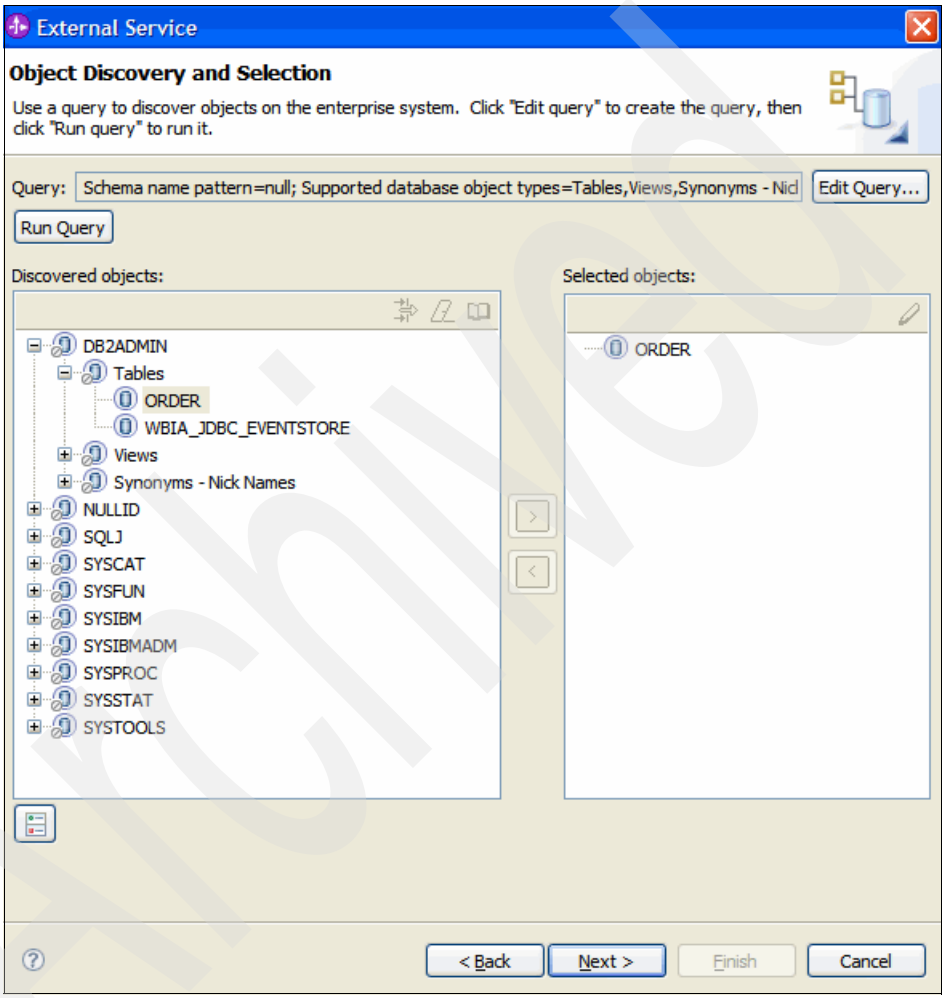


Figure 6-13 External Service – Object Discovery and Selection window

9. In the Configure Composite Properties window, click **Next**.

10. In the External Service – Service Generation and Deployment Configuration window (Figure 6-14):

    a. For J2C Authentication Data Entry, type `<NodeName>/JDBCAdapter/inbound`, where *NodeName* is the node name of the WebSphere Application Server target instance.

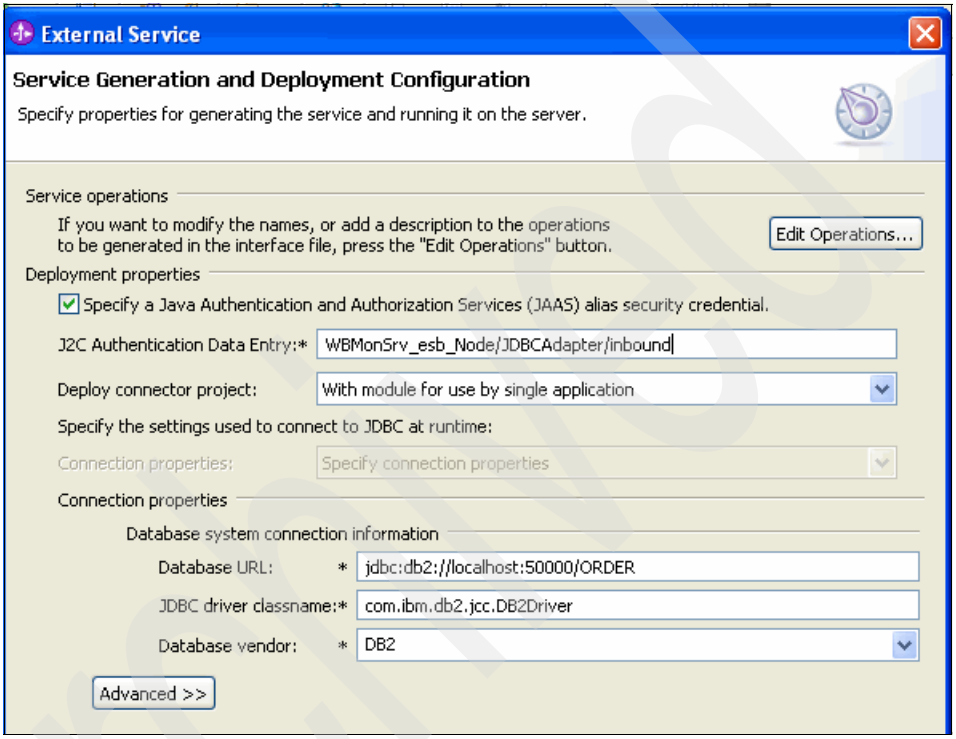


Figure 6-14 External Service – Service Generation and Deployment Configuration window

The node name for the WebSphere Business Monitor server on the WebSphere Enterprise Service Bus profile is WBMonSrv_esb_Node (Figure 6-15).

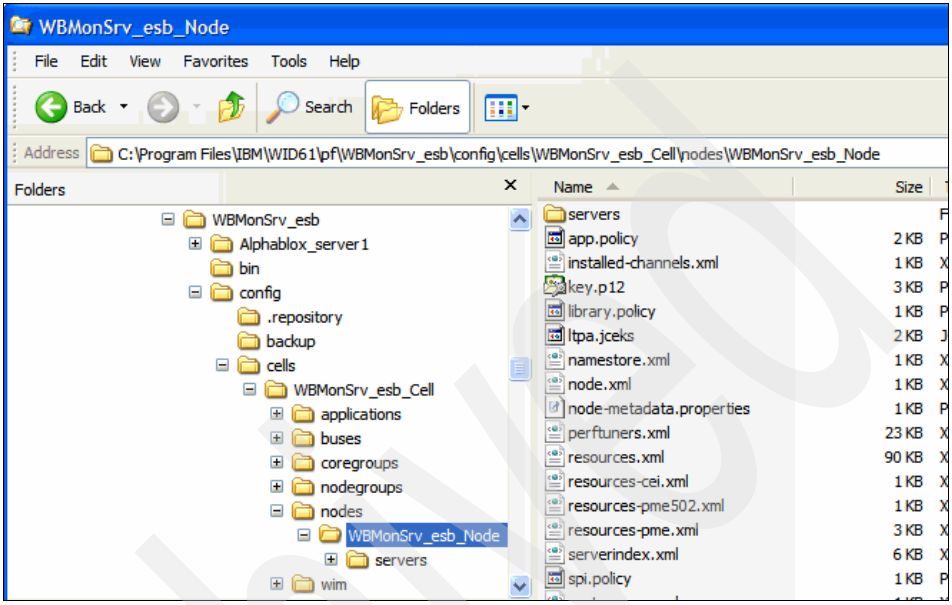


Figure 6-15 Locating the node name

Archive

b. Click the **Advanced** button and scroll down further. Expand **Advanced connection configuration** (Figure 6-16). For Datasource JNDI name, type `jdbc/DB2XA`. You create this data source later. Click **Next**.

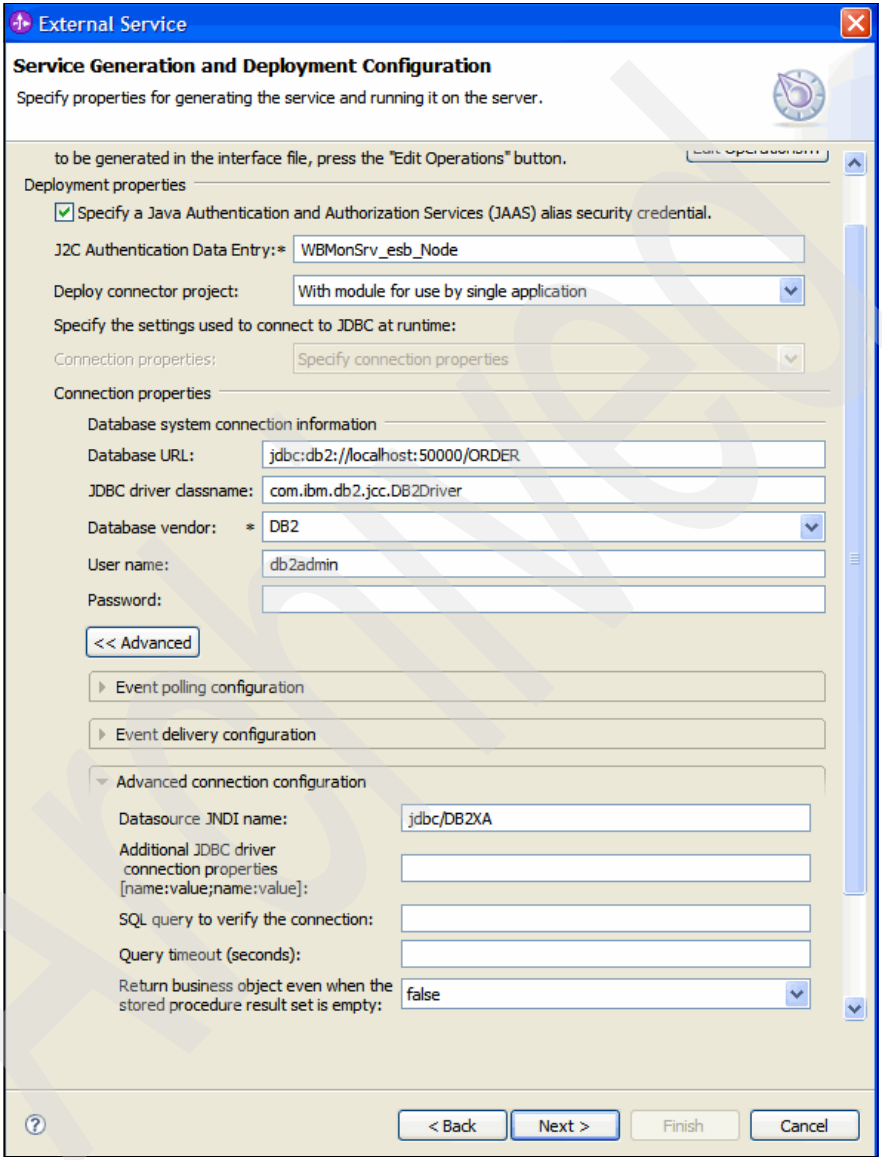


Figure 6-16 Enter datasource JNDI name

11. In the Service Location Properties window, click **Finish**.

The result of running the external service wizard is an SCA module that contains an EIS import or export. In our scenario, you observe that a JDBC Export is added to the assembly diagram.

Adding a Java component

The Assembly Diagram editor is now open. There is now a newly created SCA Export, called JDBCInboundInterface. In this section, we create an SCA component that is invoked when an event is triggered by the Order database. The SCA component in our example is a Plain Old Java Object (POJO).

Note: Our objective is to show how we can use combined ESB and Adapter technologies to receive events from a nonmanaged event source and emit events through service component monitoring. A Java component was chosen as a simple illustration of an SCA component. It prints the details of the events as they arrive from the event source (ORDER database). By using this component, we can correlate this information with the Common Base Events that are emitted from WebSphere Enterprise Service Bus.

1. In the assembly diagram (Figure 6-17), go to the palette panel. Click the **Components** tab and select **Java**. Click the assembly diagram canvas and then select the **Java** component.

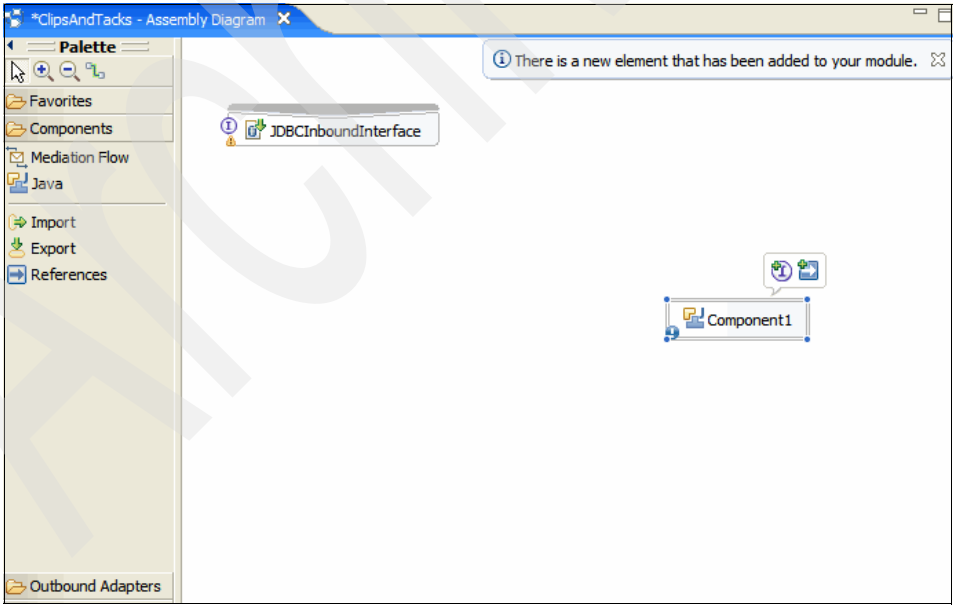


Figure 6-17 Adding a Java component

2. Click the **Properties** tab (Figure 6-18). For Name, type `OrderPOJO`.

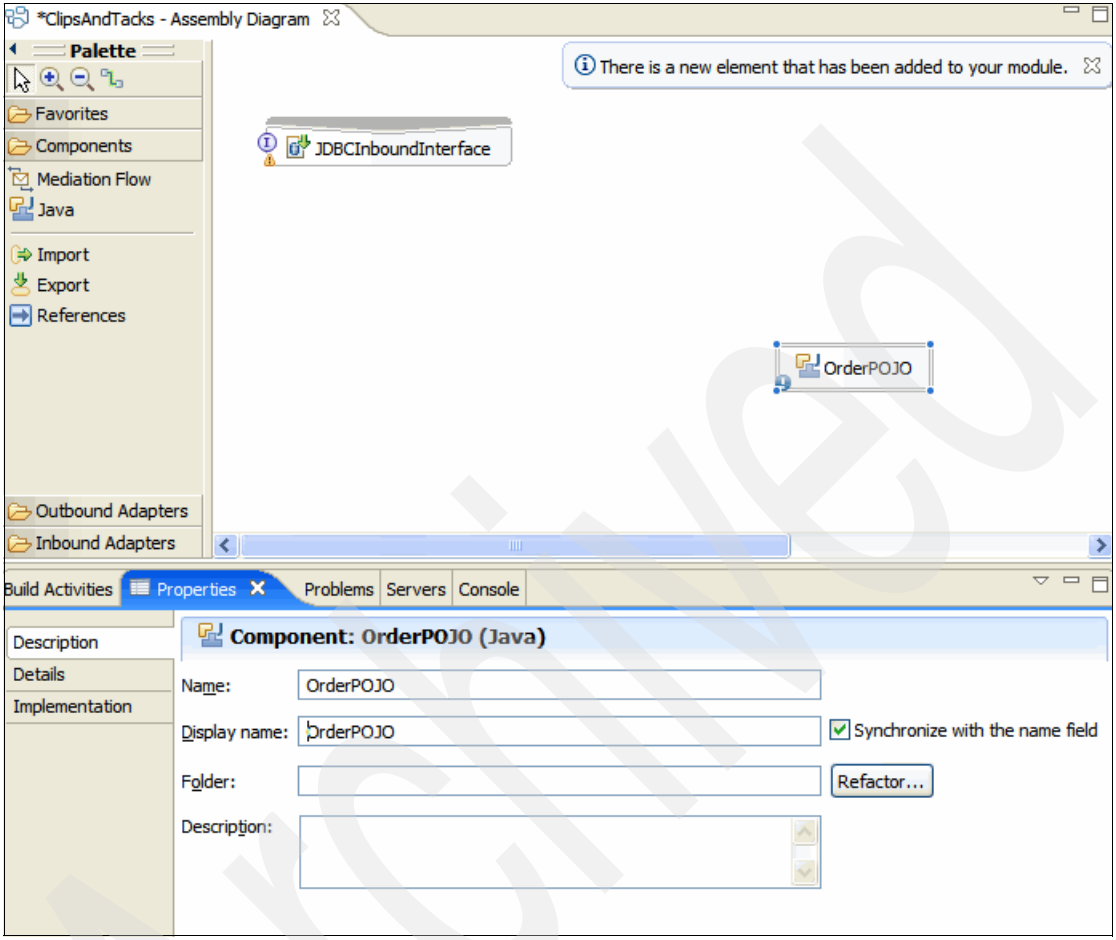


Figure 6-18 Renaming the Java component

Archived

3. Hover your cursor on the **JDBCInboundInterface** SCA Export. Drag the connector to the **OrderPOJO** Java component (Figure 6-19).

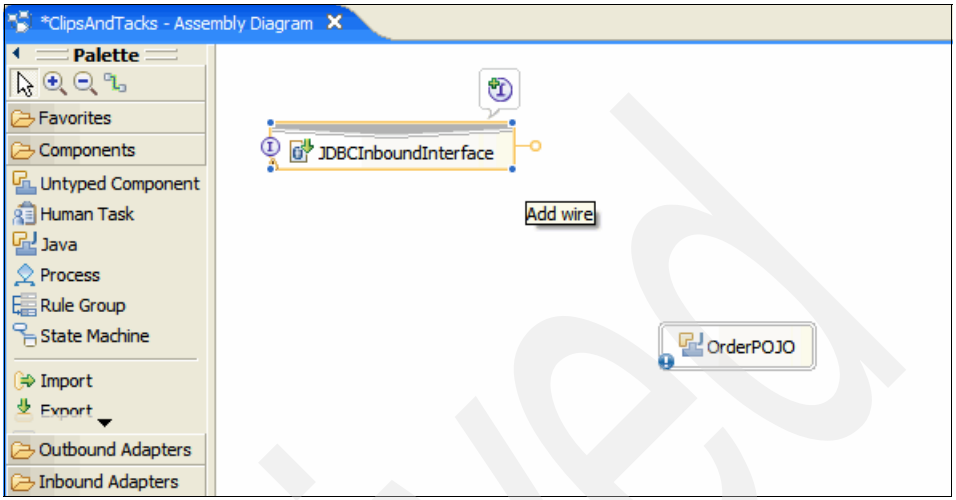


Figure 6-19 Connecting components

4. When you are prompted to add the service interface from the export to the target by the Add Wire window (Figure 6-20), click **OK**.

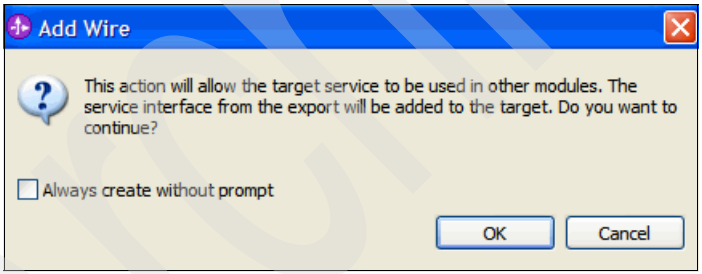


Figure 6-20 Adding a wire

Archived

5. Right-click the **OrderPOJO** SCA component and select **Generate Implementation** (Figure 6-21).

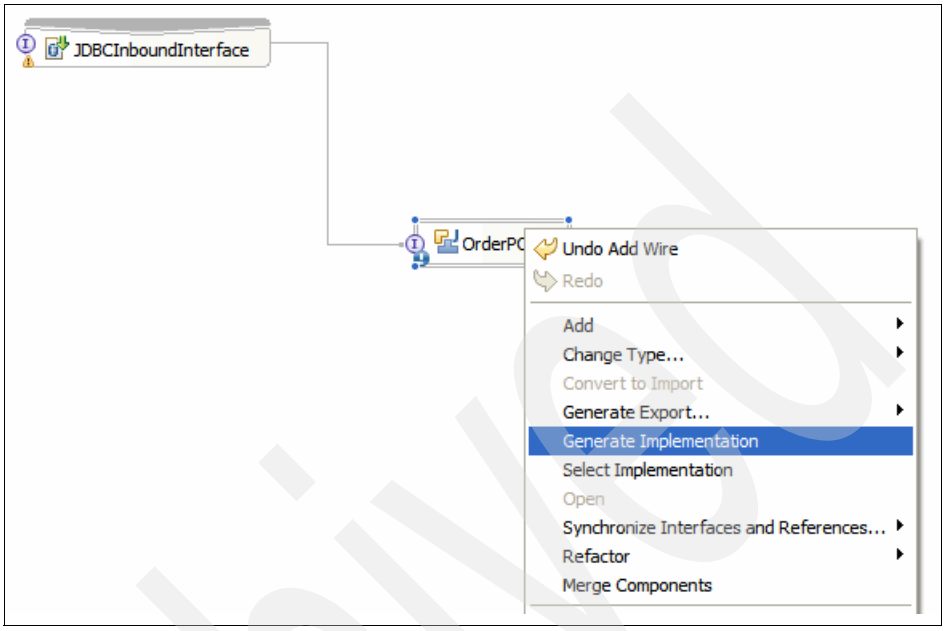


Figure 6-21 Selecting Generate Implementation

Archive

6. In the Generate Implementation window (Figure 6-22), under Select the package where the Java implementation will be generated, use the default and click **OK**. The OrderPOJO Java implementation class automatically opens.

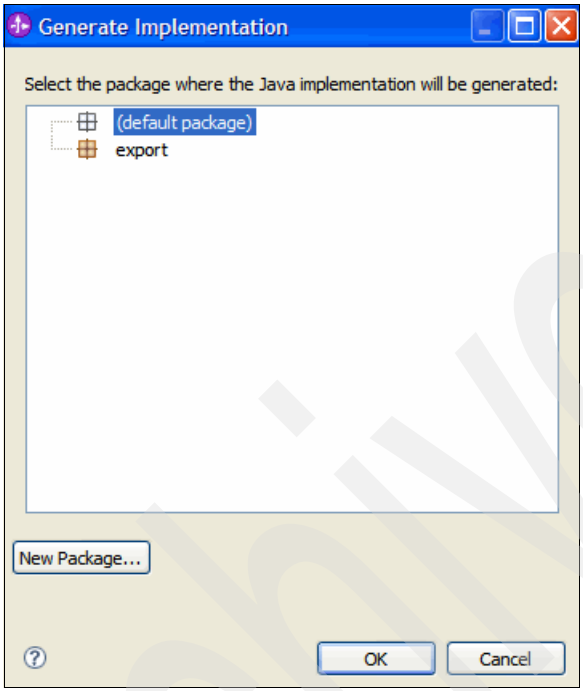


Figure 6-22 Using the default package selection

Archived

7. In the createDb2adminOrderBG method, add the code shown in Example 6-2. This code is included in the /AdapterWESB/CodeSnippet/PrintOrder.txt directory of the additional materials that are available for download for this book. See Appendix B, “Additional material” on page 801, for more information.

Service interface: In the Configure Composite Properties window of the External Service Wizard (see step 9 on page 267), you can specify the operations for the selected business object. In our scenario, we use the default settings of Create, Update, and Delete. Therefore, all three associated operations are generated in the JDBCInboundInterface interface.

When wiring the Java component to the JDBCInboundInterface component, you are prompted to add the service interface from the export to the target. Therefore, the generated Java skeleton class, includes the corresponding createDb2adminOrderBG, updateDb2adminOrderBG, and deleteDb2adminOrderBG methods.

Example 6-2 Code for createDb2adminOrderBG

```
//Print Order Record
DataObject bg=(DataObject)createDb2adminOrderBGInput;
DataObject bo=bg.getDataObject(“Db2adminOrder”);

System.out.println(“Create Order Starts”);
System.out.println(“Order number is: “+bo.getString(“onumber”));
System.out.println(“First name is: “+bo.getString(“fname”));
System.out.println(“Last name is: “+bo.getString(“lname”));
System.out.println(“Country is: “+bo.getString(“tctry”));
System.out.println(“Total price is: “+bo.getString(“tprice”));
System.out.println(“Order status is: “+bo.getString(“ostatus”));
System.out.println(“Create Order Ends”);
```

This Java implementation class prints the content of the events to the console to correlate the data that is received by the WebSphere Adapters with the events that are emitted by WebSphere Enterprise Service Bus.

8. Save the assembly diagram and OrderPOJO Java implementation class.

6.4.2 Configuring the server resources

To configure the server resources:

1. Click the **Server** tab (Figure 6-23). Right-click **WebSphere Business Monitor Server v6.1 on WebSphere ESB** and select **Start** to start the server.

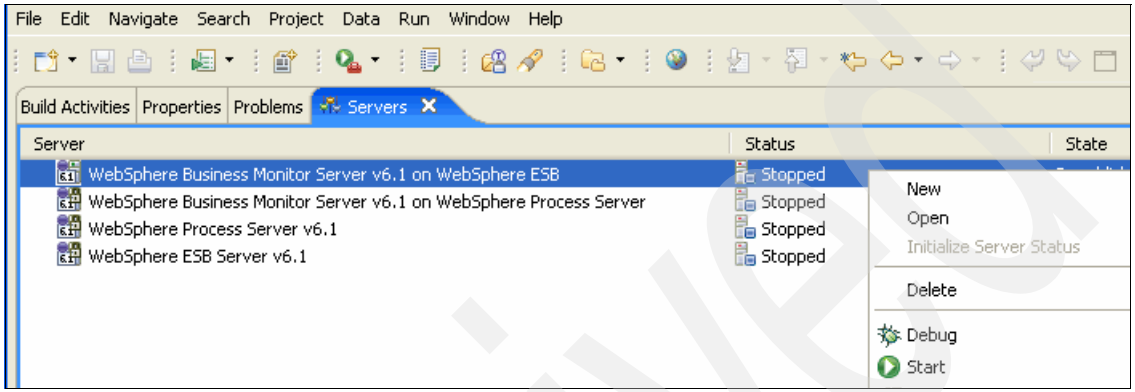


Figure 6-23 Starting the server

2. After the server status indicates that the server has started, right-click **WebSphere Business Monitor Server v6.1 on WebSphere Enterprise Service Bus** and select **Run administrative console**.
3. If prompted with a security alert (Figure 6-24), click **Yes** to proceed.

Figure 6-24 Security Alert

Archive

4. In the login panel (Figure 6-25), type the User ID and Password. For our example, we use `admin` for both.

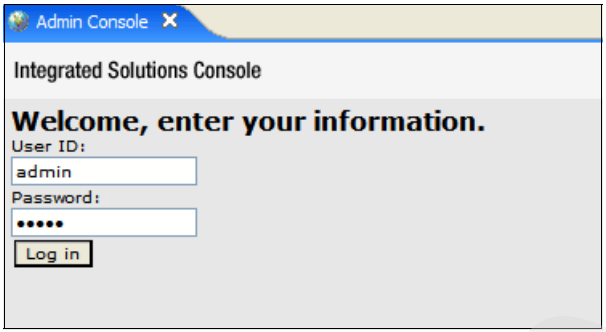


Figure 6-25 Logging in to the administrative console

Creating a J2C authentication alias

Using an authentication alias eliminates the need to store the password in clear text in an adapter configuration property where it might be visible to others. The authentication alias must be set since the adapter uses a user name and password to connect to the database. To set the authentication alias in the WebSphere Enterprise Service Bus administrative console:

1. In the Integrated Solutions Console, expand **Security** and click **Secure administration, applications, and infrastructure**.
2. Under the Authentication heading, expand **Java Authentication and Authorization Service** and select **J2C authentication data**.
3. Click **New** to create a new authentication alias.

Archived

4. In the Secure administration, applications, and infrastructure panel (Figure 6-26), enter the properties of the DB2 authentication alias using the values in Table 6-3. Click **OK**.

Table 6-3 Creating the authentication alias properties

Properties	Value
Alias	DB2
User ID	db2admin
Password	db2admin
Description	Authentication alias for DB2

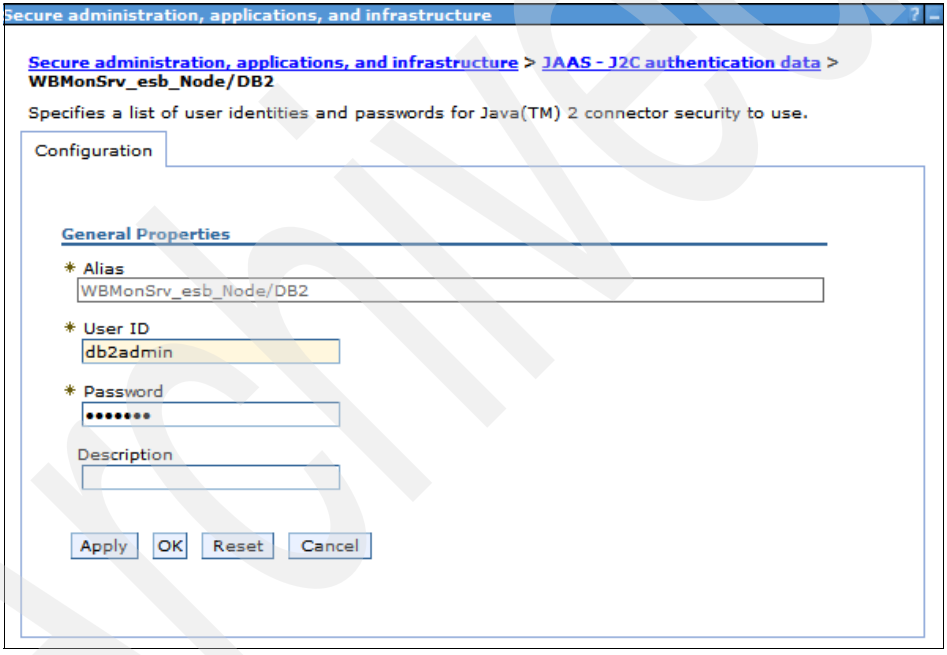


Figure 6-26 DB2 authentication alias

5. Click **New** to create another authentication alias.

Archived

6. In the Secure administration, applications, and infrastructure panel (Figure 6-27), enter the properties of the JDBCAdapter Authentication Alias using the values in Table 6-4. Click **OK**.

Table 6-4 Properties for creating the JDBCAdapter authentication alias

Properties	Value
Alias	JDBCAdapter/inbound
User ID	db2admin
Password	db2admin
Description	JDBCAdapter Authentication Alias

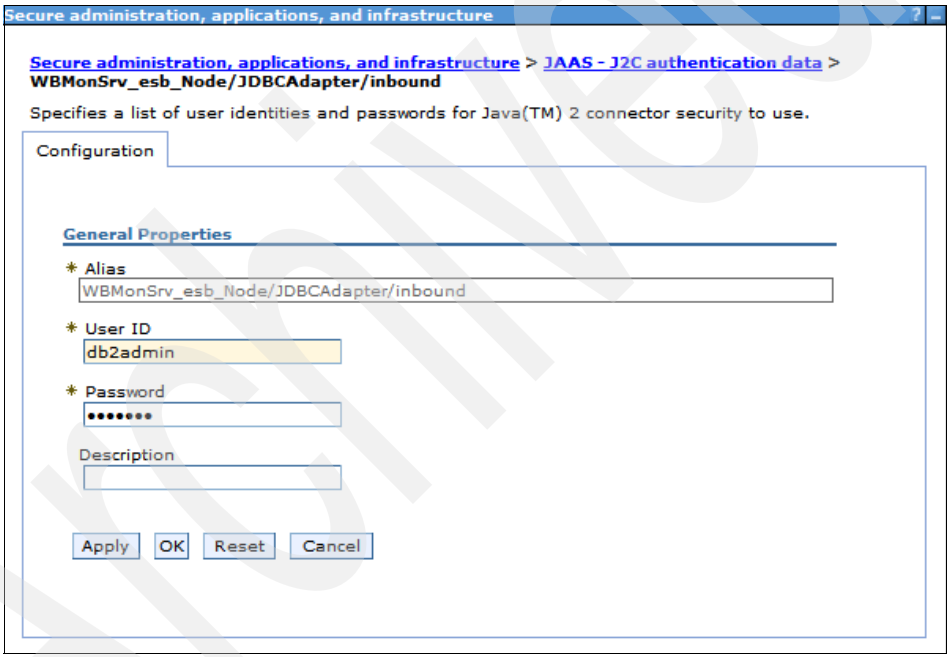


Figure 6-27 JDBCAdapter authentication alias

7. Save the changes to the master configuration.

Creating JDBC providers

By using JDBC providers, applications can interact with relational databases. The JDBC provider supplies the specific JDBC driver implementation class for access to a specific type of database. In this section, we create the JDBC provider for DB2:

1. In the Integrated Solutions Console, expand **Resources** → **JDBC** and click **JDBC Providers**. In the right pane, select the scope as server.
2. Click the **New** button.
3. In the Create a new JDBC Provider panel (Figure 6-28), create a new JDBC provider with the properties and values in Table 6-5. Click **Next**.

Table 6-5 Properties for creating a new JDBC provider

Properties	Value
Database type	DB2
Provider type	DB2 Universal JDBC Driver Provider
Implementation type	XA data source
Name	DB2 Universal JDBC Driver Provider (XA)

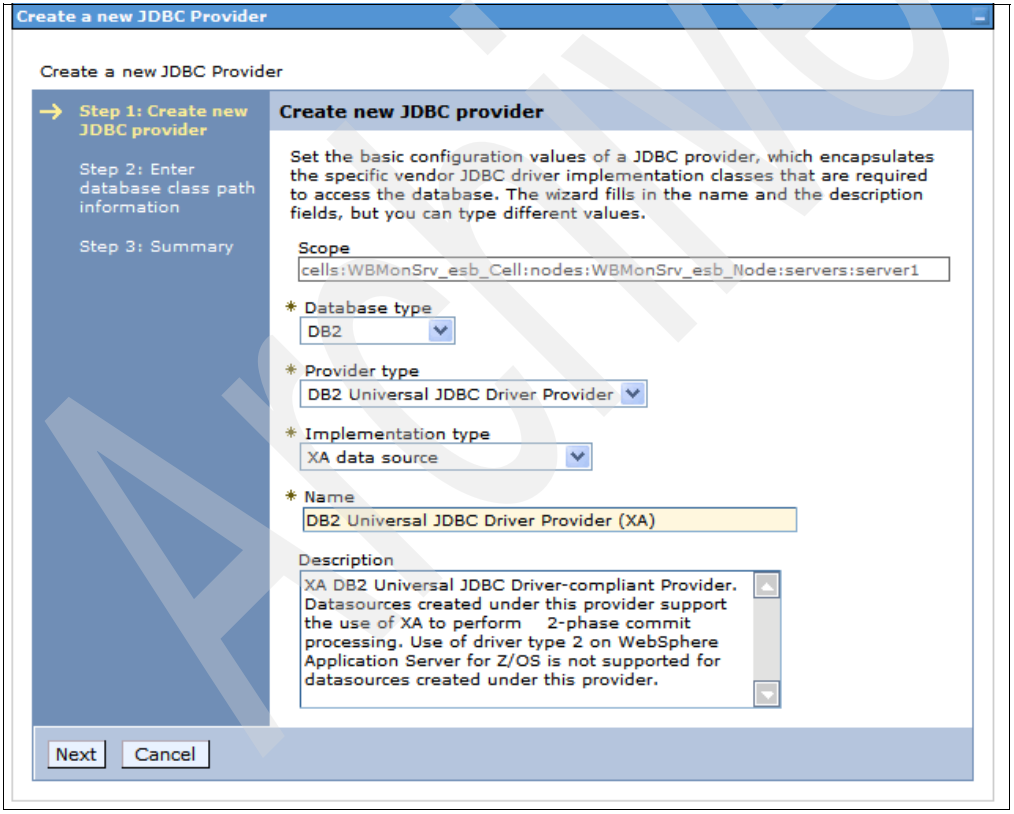


Figure 6-28 Creating a new JDBC provider

Archived

4. In the Enter database class path information panel (Figure 6-29), enter the database class path information. In our example, for ${DB2UNIVERSAL_JDBC_DRIVER_PATH} and the ${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}, we type the value `C:\Program Files\IBM\SQLLIB\java`, which is where the DB2 Java libraries are located. Click **Next**.

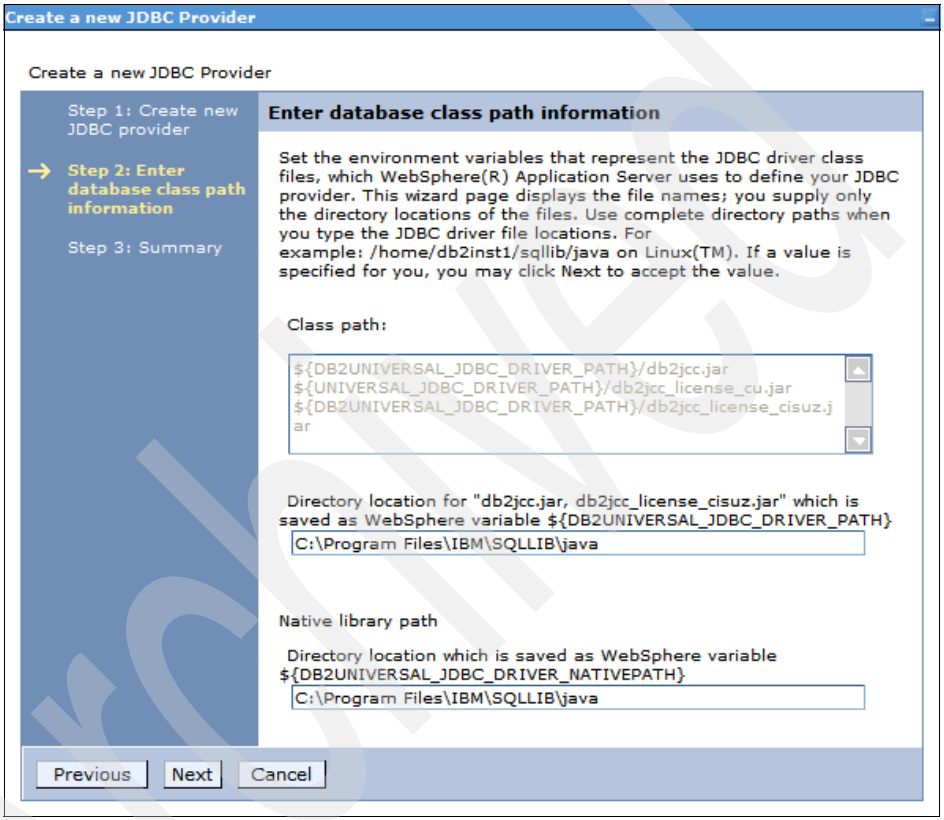


Figure 6-29 Database class path information

Archive

5. In the Summary panel (Figure 6-30), review the summary of actions. Click **Finish**.

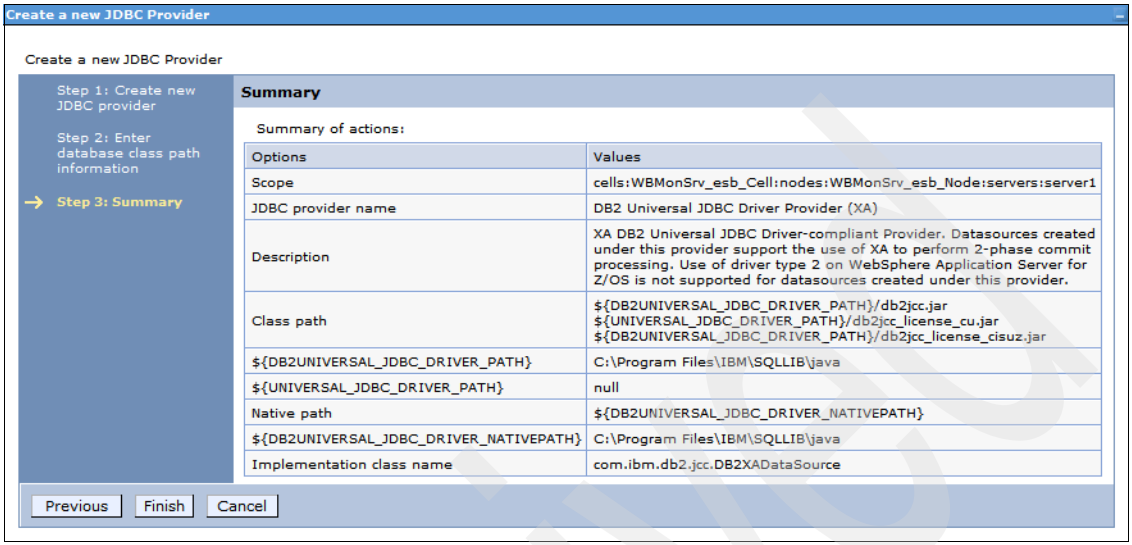


Figure 6-30 Summary of actions

6. Save changes to the master configuration.

Creating a JDBC data source

Data sources provide the link between applications and relational databases. Applications use a data source to obtain connections to a relational database. A data source is associated with a JDBC provider, which supplies the driver implementation classes that are required for JDBC connectivity with a specific type of database. In this section, we create the data source to access the order database in our scenario:

Archive

1. In the JDBC providers panel (Figure 6-31), select the newly created JDBC provider **DB2 Universal JDBC Driver Provider (XA)**.

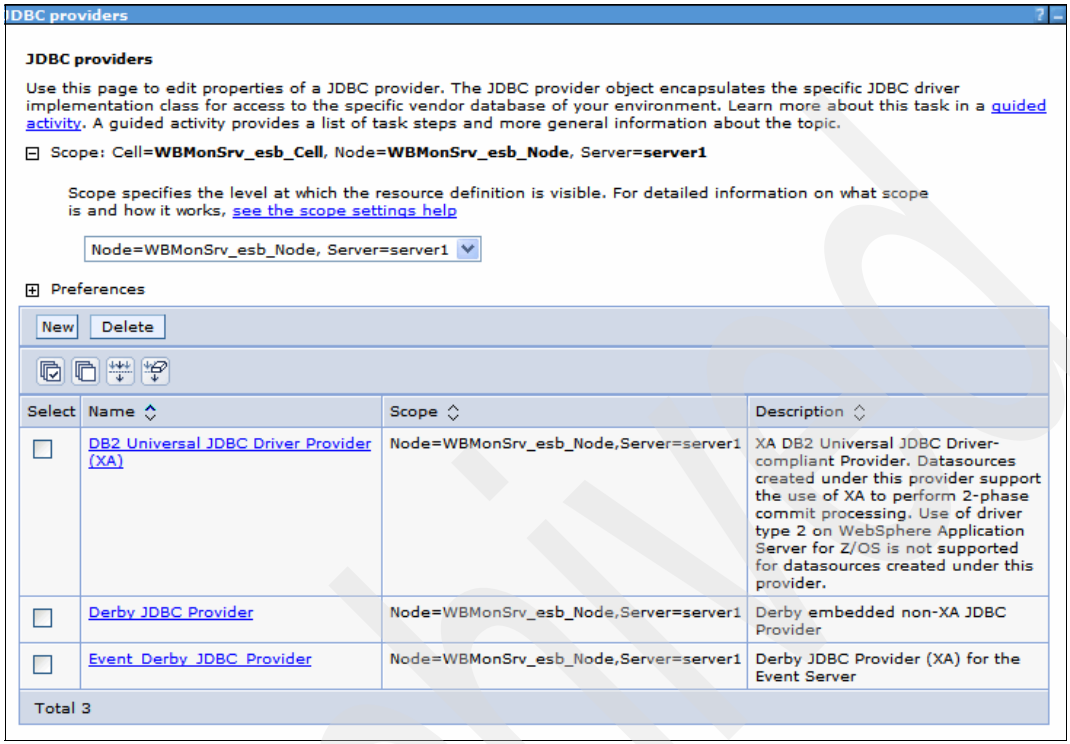


Figure 6-31 Select the newly created JDBC provider

2. On the Configuration page, under the Additional Properties, select **Data sources**.
3. Click the **New** button.

Archive

4. In the Step1: Enter basic data source information panel (Figure 6-32), for JNDI name, type `jdbc/DB2XA`. For the component-managed authentication alias, select **WBMonSrv_esb_Node/DB2**. Then click **Next**.

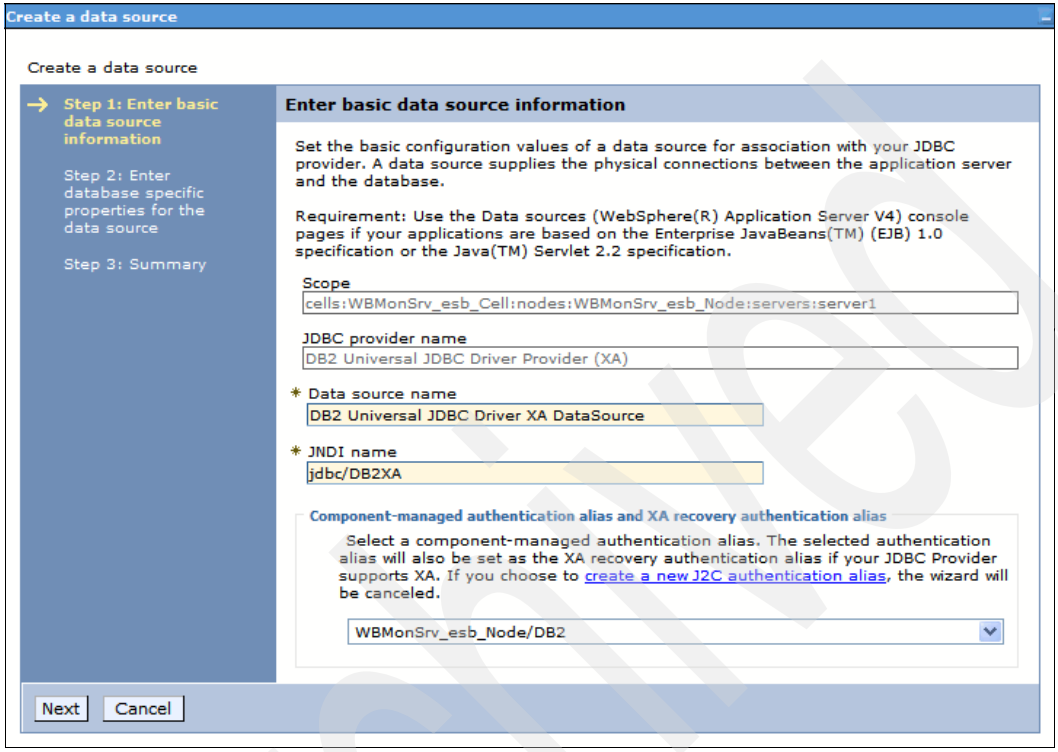


Figure 6-32 Entering the basic data source information

Archive

5. In the Step 2: Enter database specific properties for the data source panel (Figure 6-33), for Database name, type `ORDER`, and for Server name, type the host name, which in this example is `localhost`. Click **Next**.

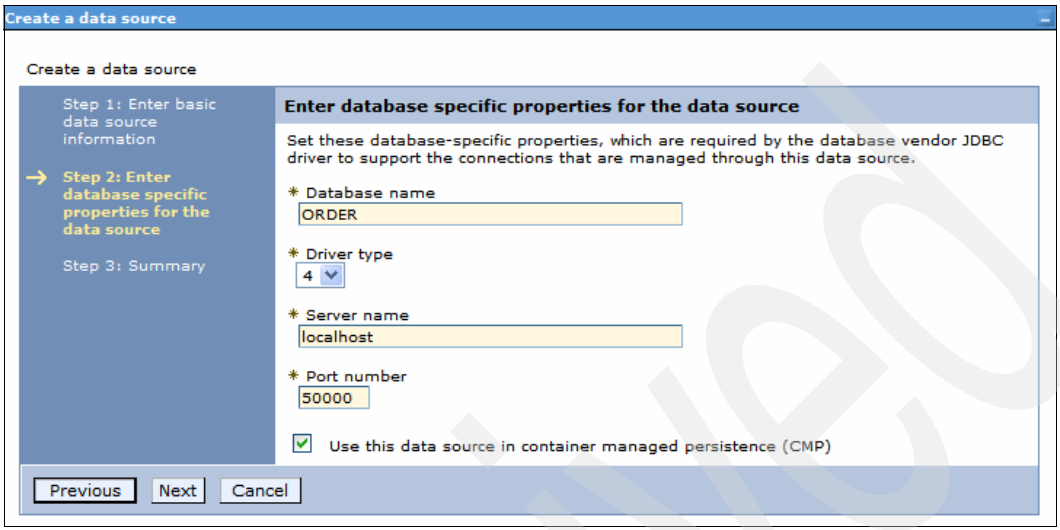


Figure 6-33 Database-specific properties for the data source

6. Review the summary of actions and click **Finish**.
7. Save the changes to the master configuration.

Archived

8. Select the newly created datasource and click **Test Connection**. You see a message that indicates the test connection operation was successful (Figure 6-34).

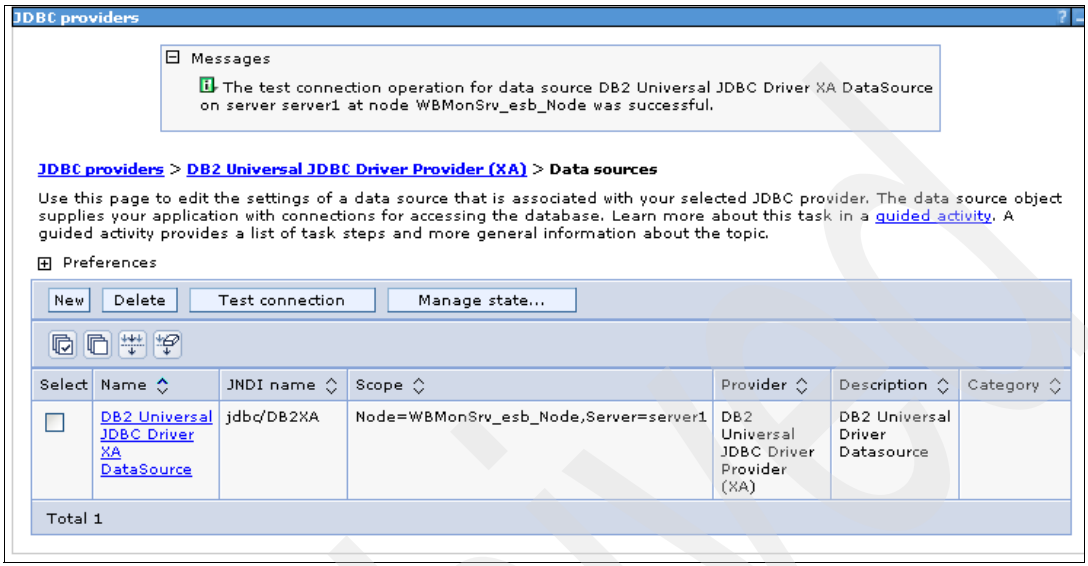


Figure 6-34 Test connection

Event service security

The CEI event services define security roles associated with a related group of function. These security roles can be used to control access to the CEI functions. For production purposes, allow only selected users to have access to the event service. However for development and testing purposes, we allow everyone to have the access to all of the CEI functions:

1. In the Integrated Solutions Console, expand **Service integration** → **Common Event Infrastructure** and click **Event service**.
2. In the right pane, under Additional Properties (Figure 6-35 on page 288), click **Map security roles to users or groups**.

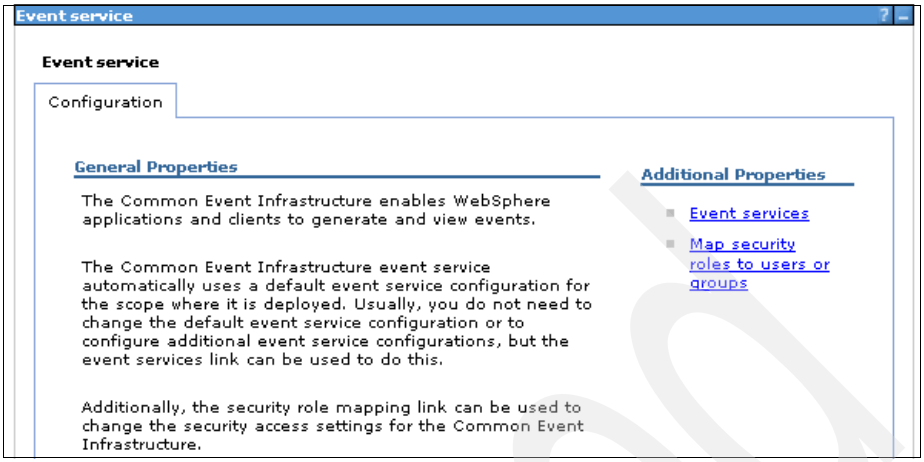


Figure 6-35 Event service

3. In the Event service panel (Figure 6-36), select everyone for all the roles listed. Click **OK**.

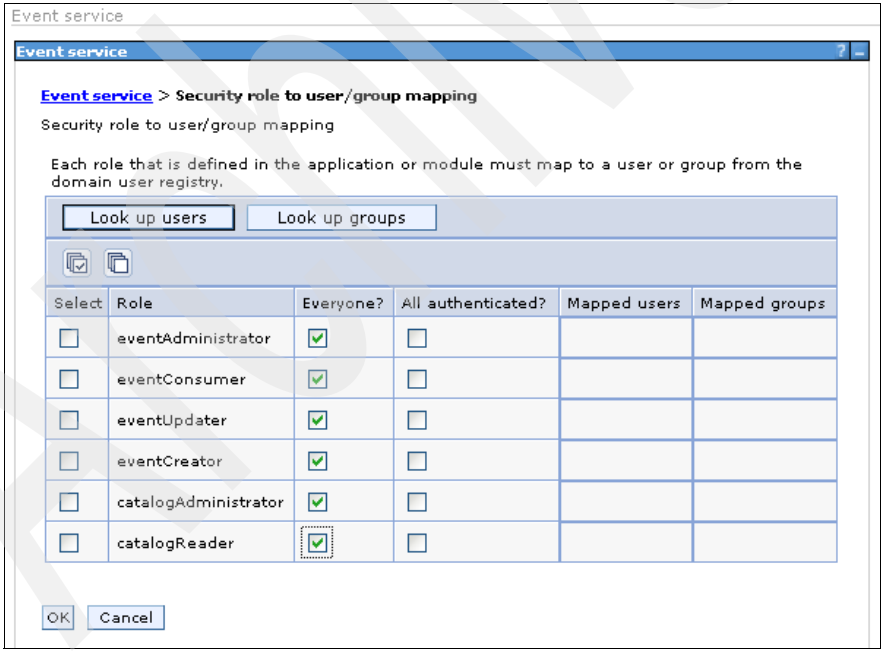


Figure 6-36 Security role to user/group mapping

4. Click the **Save** directly to the master configuration link.

5. Log out and close the Integrated Solutions Console.
6. In the Servers view of the WebSphere Integration Developer, right-click the **WebSphere Business Monitor Server v6.1 on WebSphere Enterprise Service Bus** server and select **Restart**.

6.4.3 Testing the adapter

In this section, we verify that we can receive events from the Order database in WebSphere Enterprise Service Bus:

1. In the Servers view of the WebSphere Integration Developer, right-click the **WebSphere Business Monitor Server v6.1 on WebSphere Enterprise Service Bus** server and select **Add and Remove projects**.
2. In the Add and Remove Projects window (Figure 6-37), select **ClipsAndTacksApp** and click the **Add** button. Then click **Finish**.

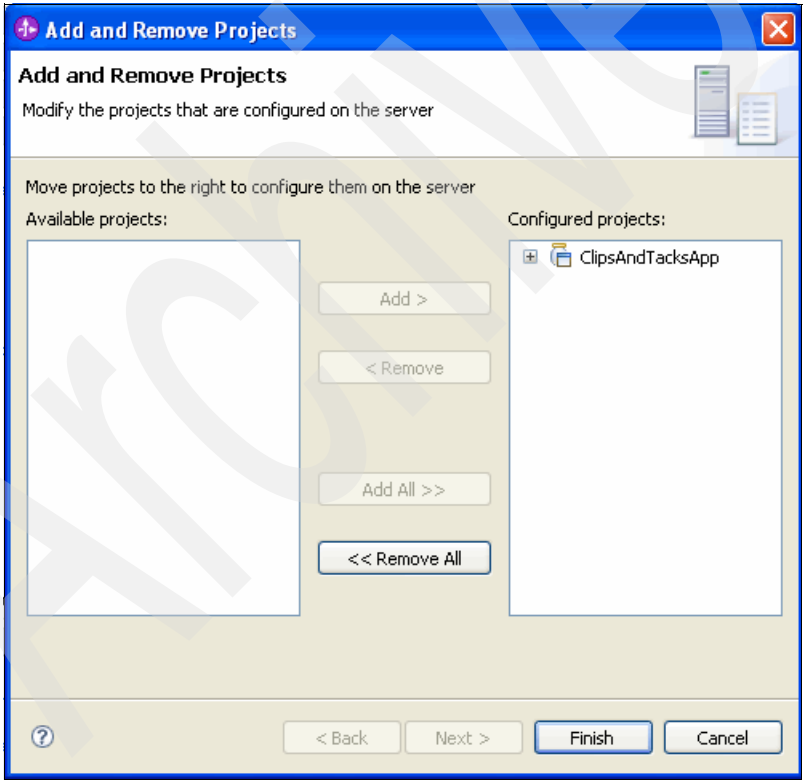


Figure 6-37 Add and Remove Projects window

Archived

3. After the ClipsAndTacksApp is successfully published (Figure 6-38), open the DB2 Command Line processor.

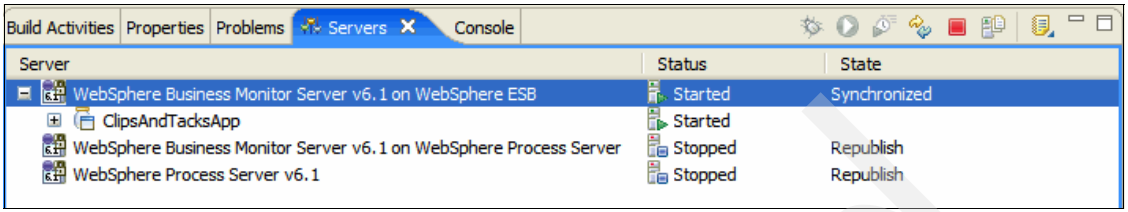


Figure 6-38 ClipsAndTacksApp started

4. We include a sample script, loadOrder.sql, under code/AdapterWESB/LoadOrder.sql, for testing the adapter. In the DB2 Command Line processor, navigate to the directory where you saved the file and enter the following command:

```
db2 -tvf LoadOrder.sql
```

Example 6-3 shows the details of the script.

Example 6-3 loadOrder.sql

```
connect to ORDER user db2admin using db2admin;
insert into ORDER values ('1', 'Tom','Fox', 'USA', '20', 'started');
insert into ORDER values ('2', 'Alan','Lim', 'SINGAPORE', '16',
'started');
insert into ORDER values ('3', 'Johanna','Ang''ani', 'GERMANY',
'50', 'started');
insert into ORDER values ('4', 'Yi','Che', 'CHINA', '17',
'started');
insert into ORDER values ('5', 'Supannee','Amnajmongkol',
'THAILAND', '30', 'started');
insert into ORDER values ('6', 'Martin','Keen', 'UK', '15',
'started');
disconnect ORDER;
```

Archived

5. If your test is successful, the console view in WebSphere Integration Developer is displayed like the example in Figure 6-39. It indicates that the JDBCInboundInterface component detected changes to the ORDER database and that the OrderPOJO component received and sent the outputs correctly to the console.

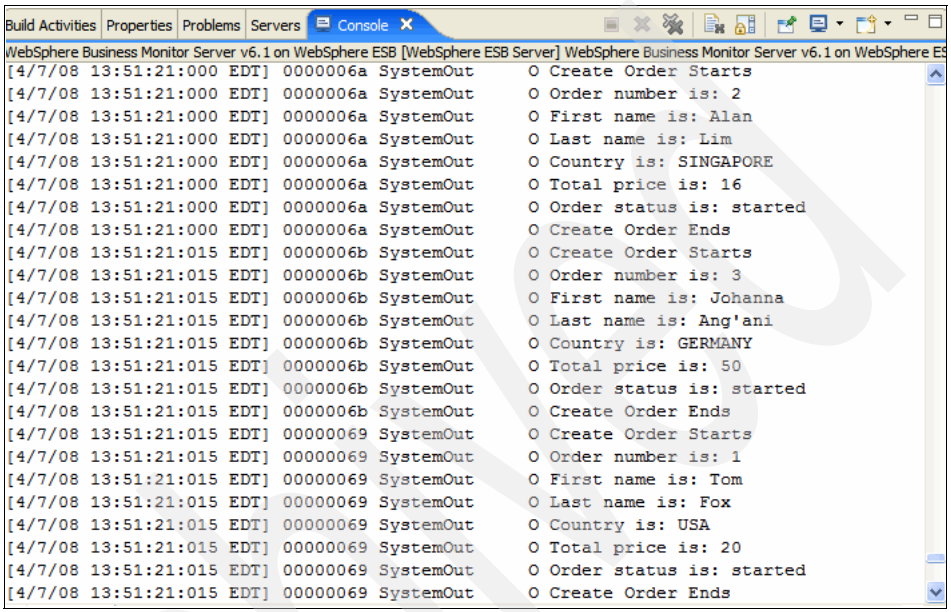


Figure 6-39 Adapter success

6. After you complete the testing, connect to the database again and delete the records from the tables by entering the commands as shown in Example 6-4:

Example 6-4 Deleting records from database tables

```
db2 connect to ORDER user db2admin using db2admin
db2 delete from ORDER
db2 disconnect ORDER
```

6.4.4 Enabling event generation

In WebSphere Integration Developer, the event monitor is the designated tool for selecting events for generation. The event monitor can be opened from the Properties view. It features a simple user interface so that you can easily select and manage events. In this section, we select the events that are to be generated from the service components in our example:

1. In the assembly diagram of WebSphere Integration Developer, select the **OrderPOJO** component.
2. In the Properties view:
   a. Click the **Details** tab (Figure 6-40). Expand **Interfaces** → **JDBCInboundInterface** and select **createDb2adminOrderBG**.
   b. Click the **Event Monitor** tab. For Monitor, select **All** and for Event Content, select **Full**.

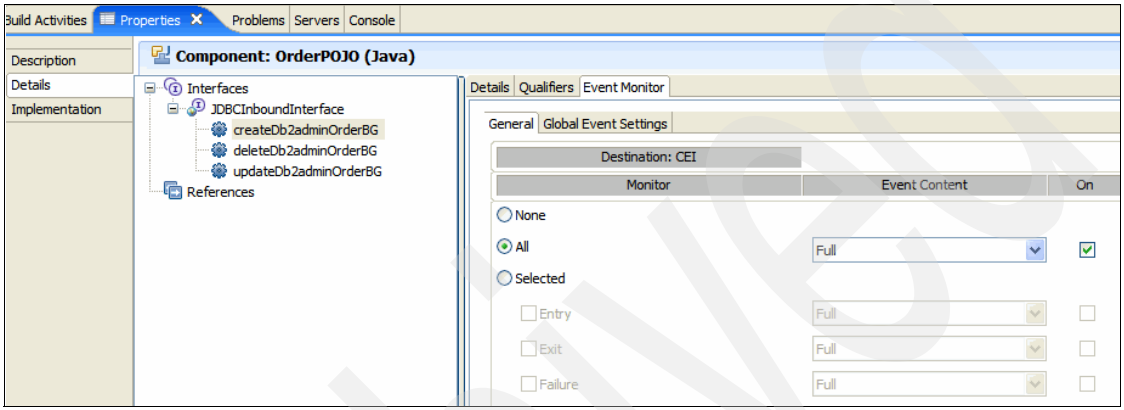


Figure 6-40 General event settings

3. Click the **Global Event Settings** tab (Figure 6-41). The default event format is the WebSphere Business Monitor v6.1 format.

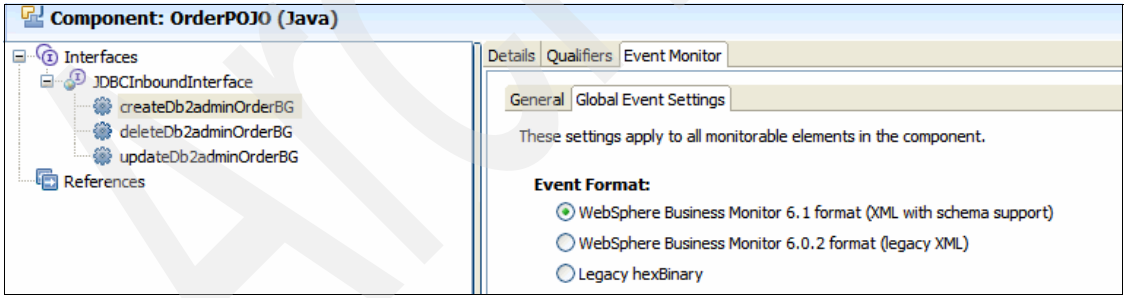


Figure 6-41 Global Event Settings

Event properties for individual component elements: In the event monitor, you can specify event properties for individual component elements that you selected in a component editor. For example, you can generate all events, selected events, or no events. You can also specify other event properties, such as the content level and transaction behavior.

4. Save the assembly diagram.
5. Click the **Server** tab. Right-click the **WebSphere Business Monitor v6.1 on WebSphere ESB server** and select **Publish**.

6.4.5 Validation

Test that the Common Base Event events are emitted correctly. Make sure that the WebSphere Business Monitor v6.1 on WebSphere ESB server is running before proceeding.

1. Verify that the order records are deleted. In the DB2 Command Line Processor, run the **`LoadOrder.sql`** script (Example 6-3 on page 290).
2. Click the **Server** tab. Right-click the **WebSphere Business Monitor v6.1 on WebSphere ESB server** and select **Common Base Event Browser**.
3. In the Common Base Event Browser, if prompted, log in to the Common Base Event browser. The default user name and password is `admin`.
4. In the Common Base Event Browser, you see a number of events that have been generated. Click the **All Events** link.
5. In the All Events view (Figure 6-42), go to the most recent events and select one of the events. Click the hyperlink under the Creation Time column.

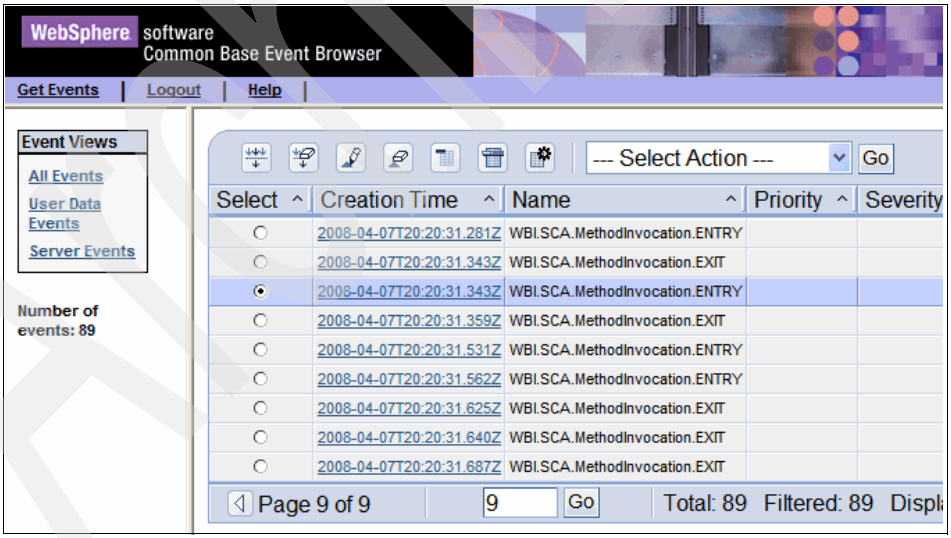


Figure 6-42 Recent events

6. In the Event Data Frame, scroll down and click the value of the **wbi:event** property.

7. Review the content of the selected event, which is similar to the example in Figure 6-43. Correlate this with the printout by the Java component.

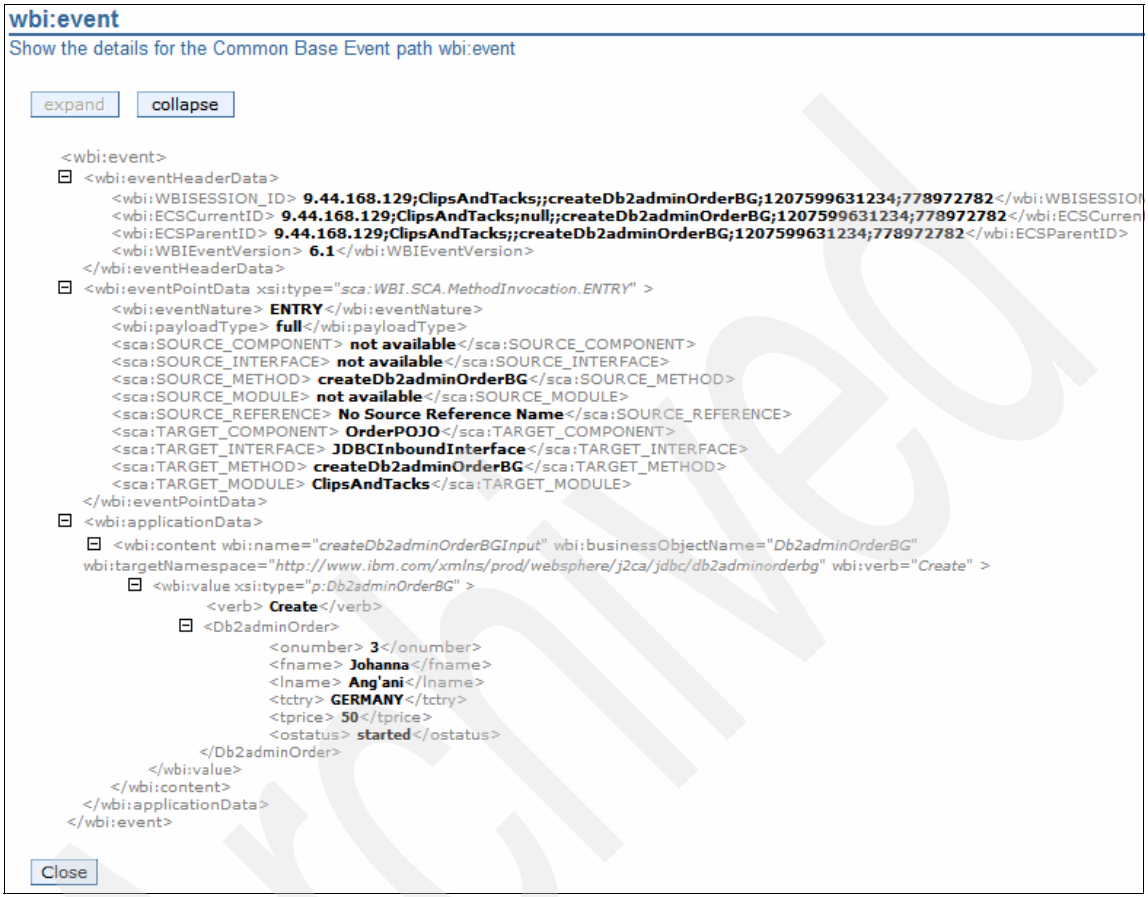


Figure 6-43 Sample event

8. When you have completed testing, connect to the database again and delete the records from the tables (Example 6-4 on page 291).
9. Close the Common Base Event Browser and return to the WebSphere Integration Developer workspace.
10. Go back to the **Server** tab. Right-click the **WebSphere Business Monitor Server v6.1 on WebSphere Enterprise Service Bus** server and select **Add and Remove projects**.
11. Select **ClipsAndTacksApp** and click the **Remove** button.

Archived

6.4.6 Creating the monitor model

By using the WebSphere Business Monitor Toolkit, you can generate a monitor model automatically from WebSphere Integration Developer without creating a monitor model from scratch.

Generating the monitor model

To generate the monitor model:

1. In WebSphere Integration Developer, in the Business Integration view, right-click the **ClipsAndTacks** module and select **Monitor Tools** → **Generate Monitor Model**.
2. Click the **New Project** button.
3. In the New Business Monitoring Project window (Figure 6-44), for Project name, type `ClipsAndTacksBM` and click **Finish**.

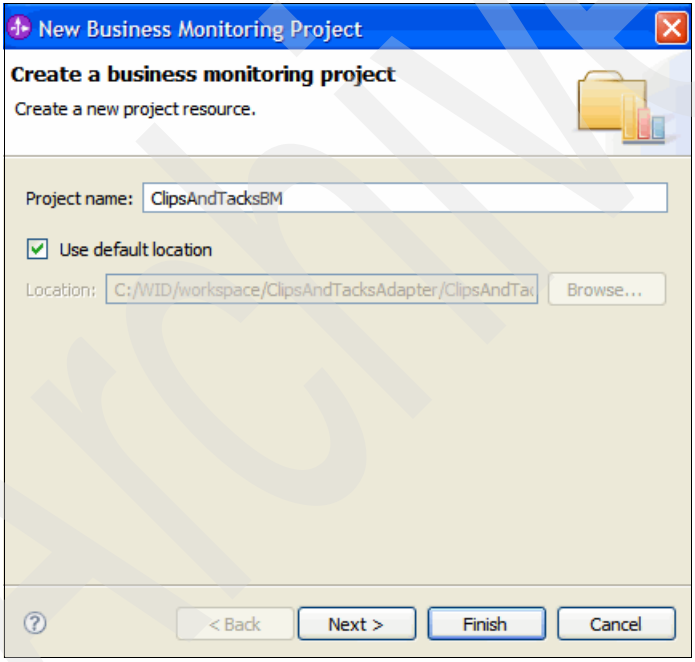


Figure 6-44 Creating a new business monitoring project

Archived

4. In the Generate Monitor Model window (Figure 6-45), for Target monitor model name, type `ClipsAndTacksMM`. Click **Next**.

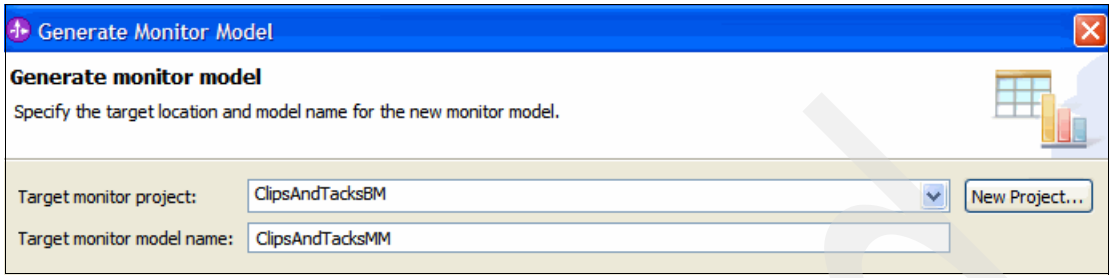


Figure 6-45 Specifying the target location of monitor model

5. In the Generate Monitor Model – Choose what to monitor window (Figure 6-46):
   a. Select **OrderPOJO** → **JDBCInboundInterface. createDb2adminOrderBG**.
   b. Click the **Emitted Events** tab.
   c. Select all the emitted events. We select all the events in this scenario because we are interested in monitoring the entry, exit, and failure events for the createDb2adminOrderBG operation.
   d. Click **Next**.

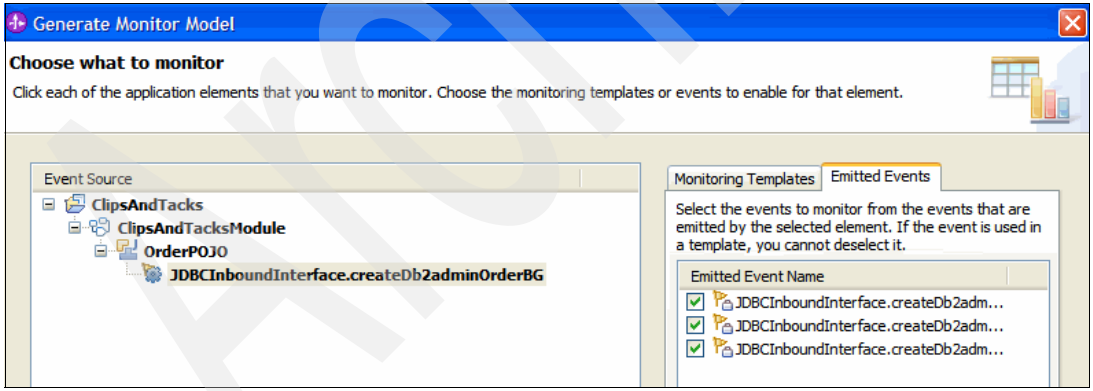


Figure 6-46 Selecting all emitted events

6. In the Generate Monitor Model – Choose how to monitor window, click **Next**.

7. In the Generate Monitor Model – Preview the monitor model window (Figure 6-47), click **Expand All** to see the elements of the Monitor Model. Click **Finish**.

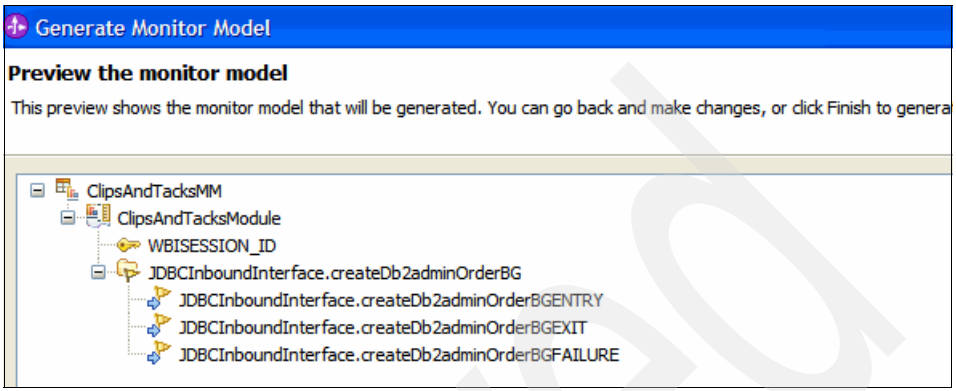


Figure 6-47 Previewing the monitor model

8. When prompted by the Switch perspectives message (Figure 6-48), click **Yes** to go to the Business Monitoring perspective. Then click **No** when prompted to launch the Get Started information.

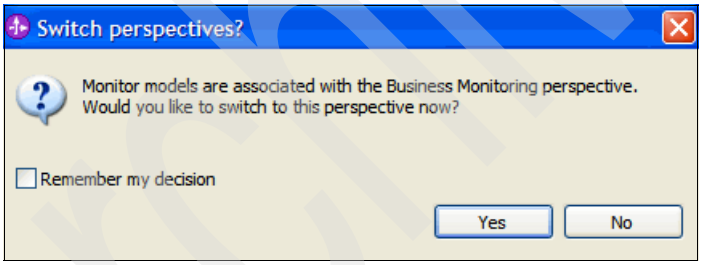


Figure 6-48 Switching to the Business Monitoring perspective

Augmenting a seed monitor model

The monitor toolkit generates a monitor model that serves as a base. In this section, we augment the base model with additional business measures.

Archived

Creating a metric

To create the Order Received metric:

1. In the Monitor Details Model ClipsAndTaskMM editor (Figure 6-49), right-click **ClipsAndTacksModule** and select **New** → **Metric**.

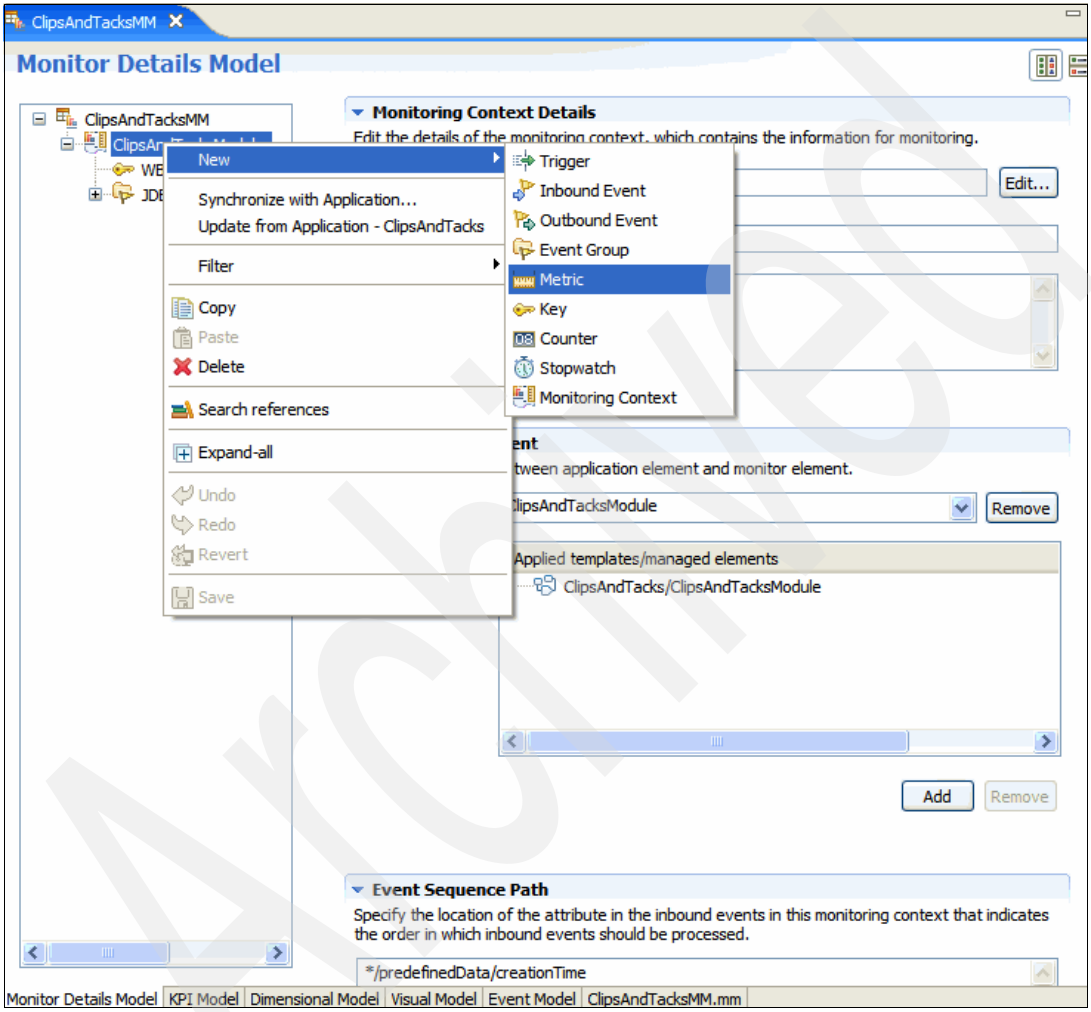


Figure 6-49 Creating a new metric

Archived

2. In the Create New Metric window (Figure 6-50), for Name, type `Order Received`. Click **OK**.

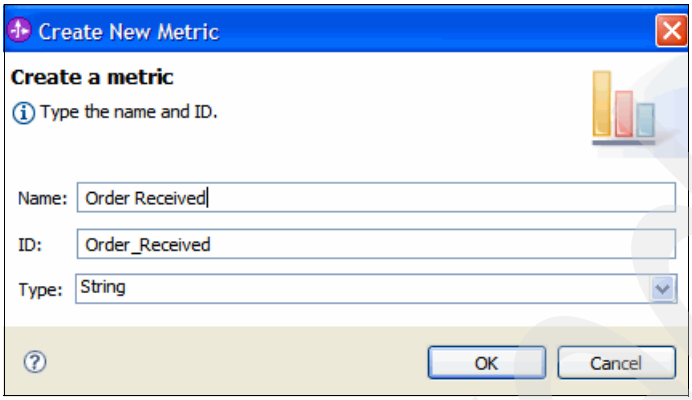


Figure 6-50 Creating an Order Received metric

3. In the right pane, under Metric Value Expressions, click **Add**.
4. Click in the space under Expression (Figure 6-51) and click the **...** button.

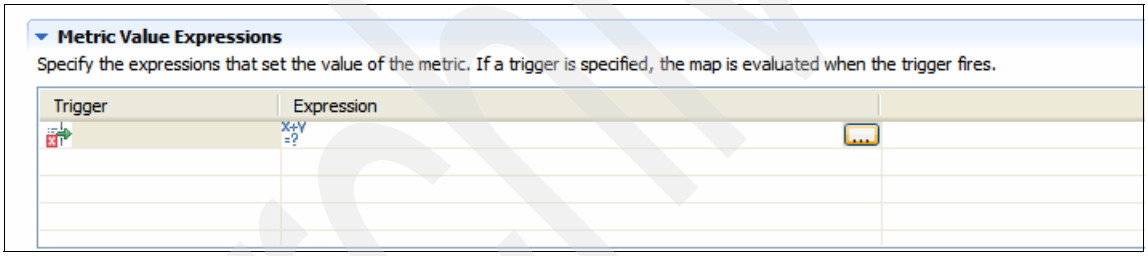


Figure 6-51 Specifying the Metric Value Expression

Archive

5. In the Expression Dialog (Figure 6-52), press Ctrl+Spacebar. Select **ClipsAndTacksMM** → **ClipsAndTacksModule** → **JDBCInboundInterface.createDb2adminOrderBGENTRY** → **createDb2adminOrderBGInput** → **Db2adminOrder** → **ostatus : string**. Then click **OK**.

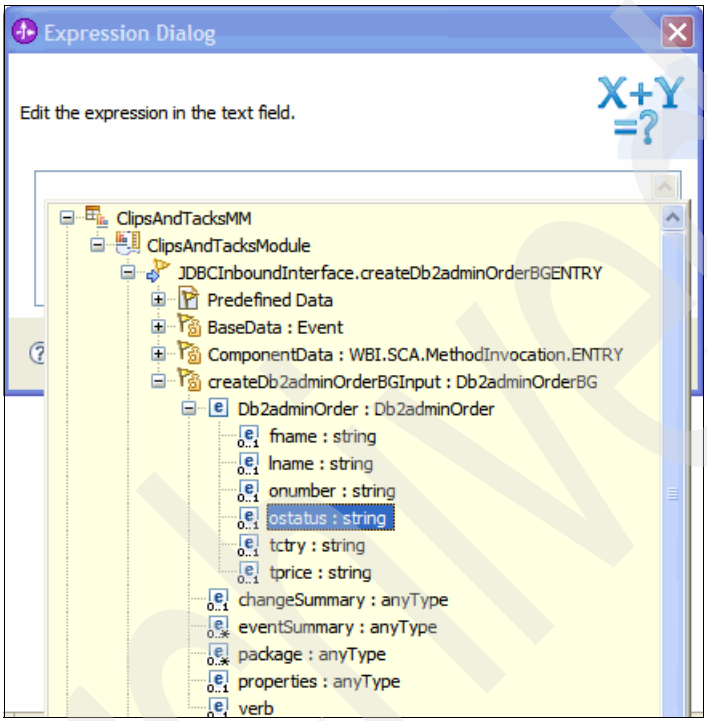


Figure 6-52 Selecting the Order Status

6. Figure 6-53 shows the completed Metric Value Expression. Save your work.

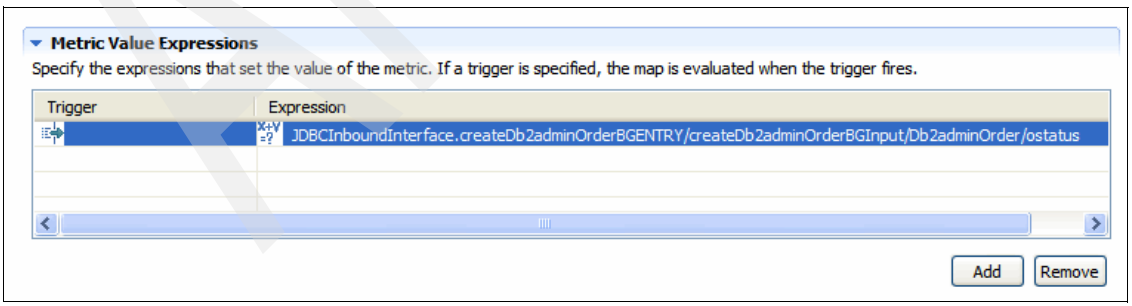


Figure 6-53 Completed Metric Value expression

Creating a KPI

Create the KPI to measure the number of orders received against a given target:

1. Click the **KPI Model** tab (Figure 6-54). Right-click **ClipsAndTacksMM** and select **New** → **KPI Context**.

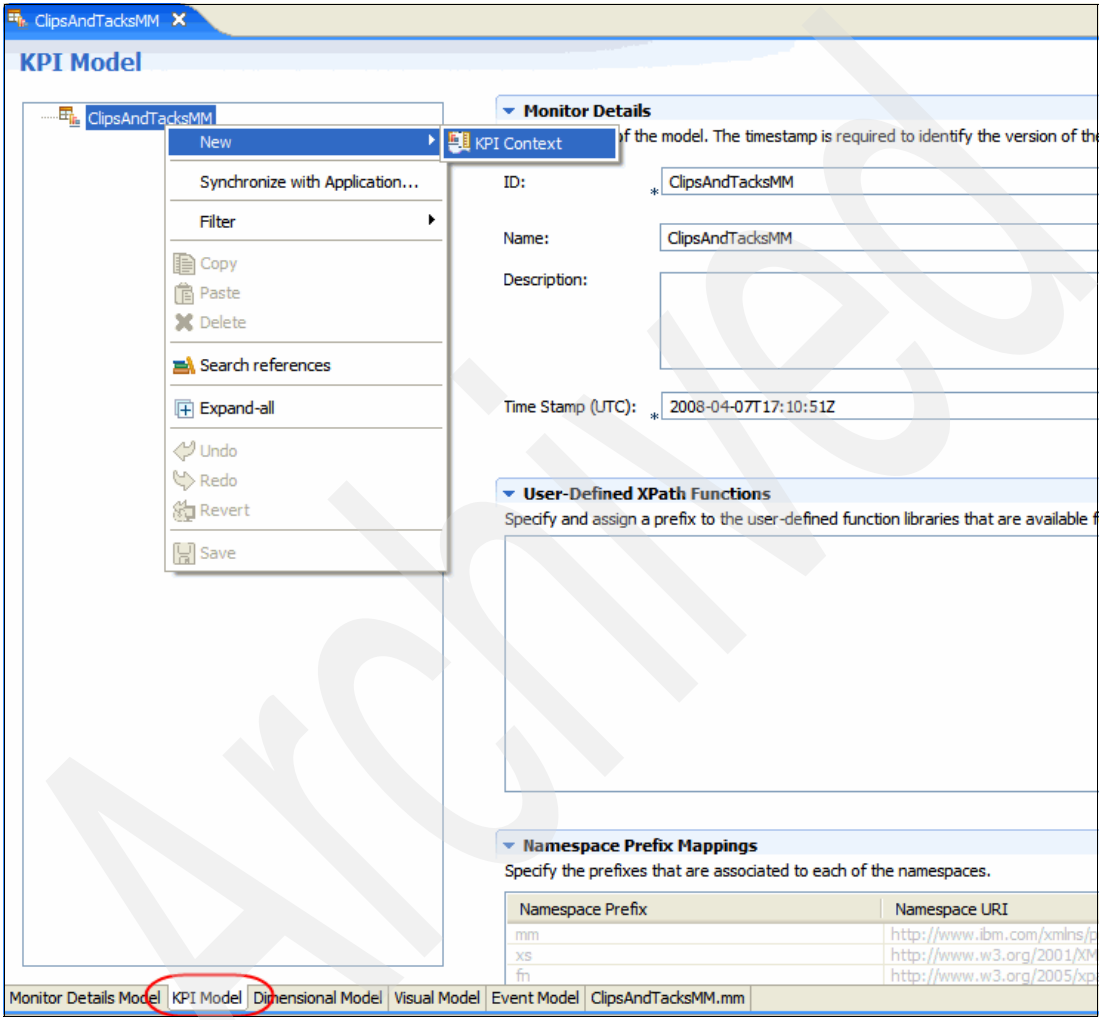


Figure 6-54 KPI Model tab

Archived

2. In the Create New KPI Context window (Figure 6-55), for Name, type `ClipsAndTacksKPIContext`. Click **OK**.

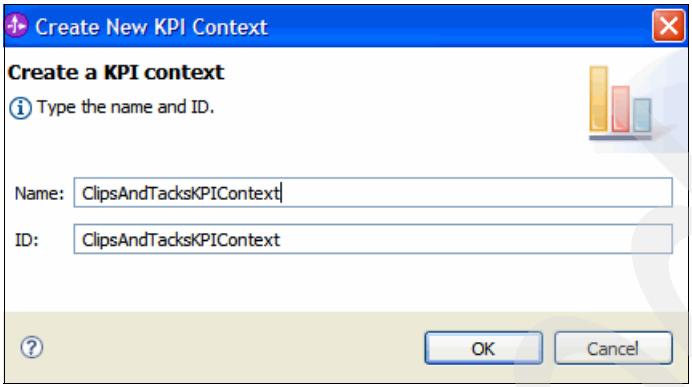


Figure 6-55 Creating a KPI Context

3. In the KPI Model panel (Figure 6-56), right-click the newly created **ClipsAndTacksKPIContext** and select **New** → **KPI**.

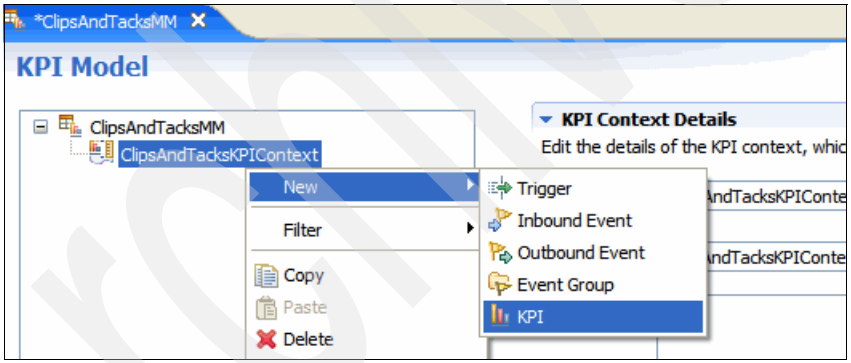


Figure 6-56 Creating a new KPI

Archived

4. In the Create New KPI window (Figure 6-56), for Name, type `No. of Orders Received`. Click **OK**.

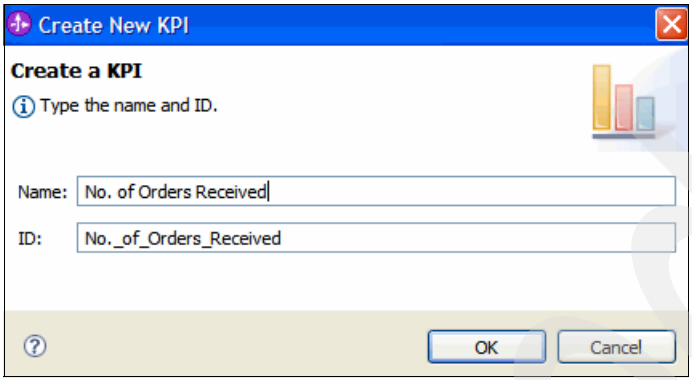


Figure 6-57 Naming the KPI No. of Orders Received

5. In the right pane, under KPI Target and Ranges (Figure 6-58), in the Target field, type `5`. In the Ranges field, click **Add**.

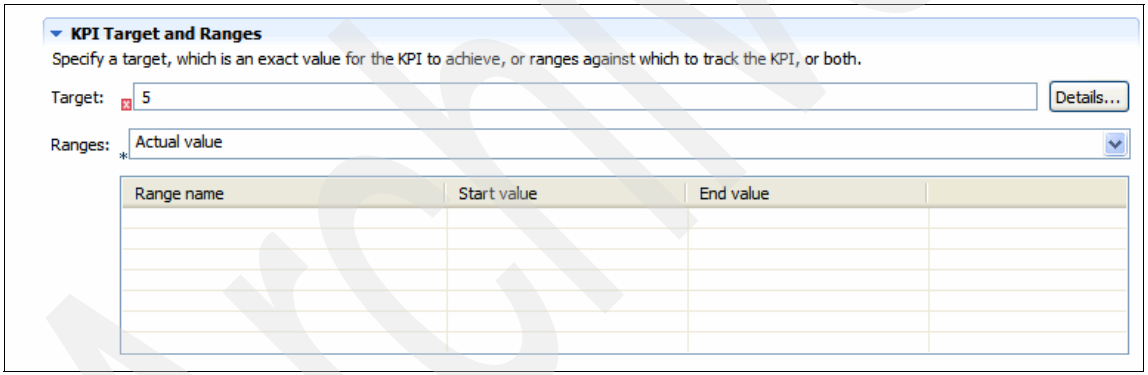


Figure 6-58 KPI target value

Archived

6. In the Add Range window (Figure 6-59), for Name and ID, type `Low` and click **OK**.

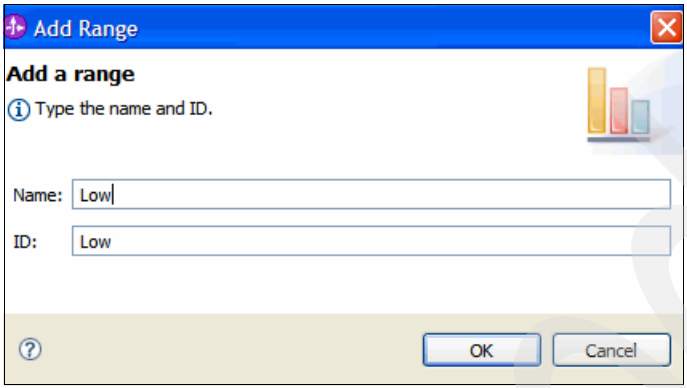


Figure 6-59 Add Low Range

7. Under KPI Target and Ranges (Figure 6-60), for Start value of Low Range, type `0` and for End value, type `3`. Then click **Add**.

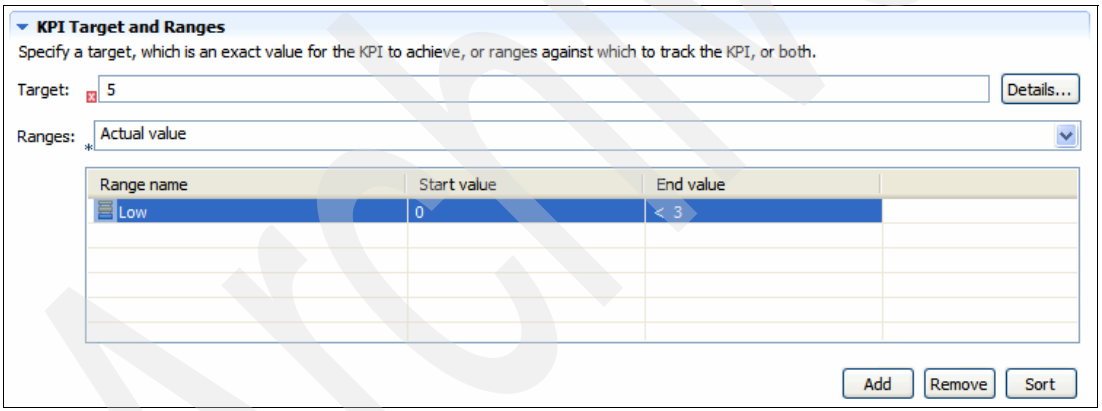


Figure 6-60 Low Range values

Archived

8. Do the same to add the Satisfactory and Excellent Ranges as in Table 6-6. Figure 6-61 shows the final result.

Table 6-6 No. of Orders KPI target range

Range name	Start value	End value
Low	0	3
Satisfactory	3	7
Excellent	7	10

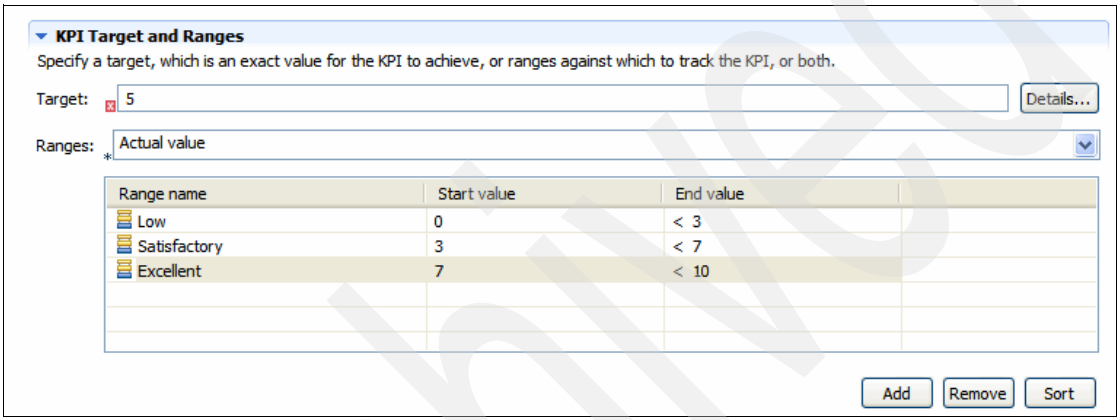


Figure 6-61 Completed Range values

9. Scroll down to the KPI Definition section (Figure 6-62). Under the KPI Value heading, select **Base this KPI on a metric and an aggregate function**.

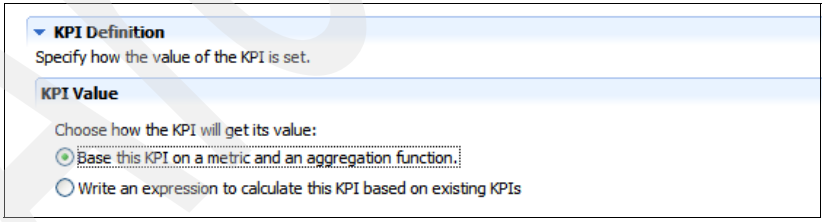


Figure 6-62 Basing the KPI on a metric

10. Define the function for calculating the number of orders received. Under the KPI Details heading (Figure 6-63), for Monitoring context, select **ClipsAndTacksModule**, for Metric, select **Order Received**, and for Aggregation function, select **Count**.

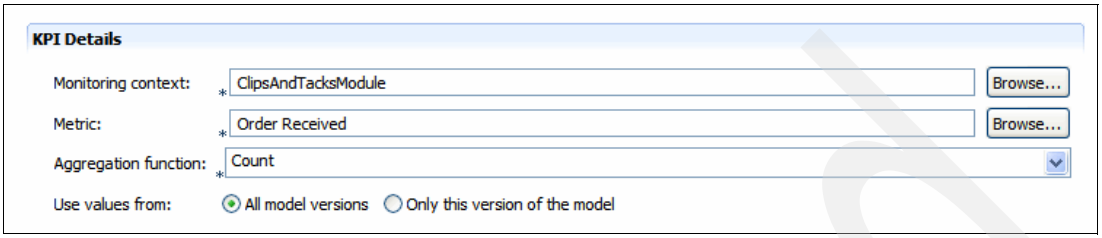


Figure 6-63 Completed KPI Details section

11. Save your work and close the monitor model.
12. Rebuild the project from scratch to resolve any broken links before deploying the project:
    a. On the menu bar, select **Project** → **Clean**.
    b. Select **Clean All Projects** and click **OK**.

Generating and deploying the Monitor J2EE project

To generate and deploy the Monitor J2EE project:

1. In the Project Explorer view, right-click **ClipsAndTacksMM.mm** and select **Generate Monitor J2EE Project**.
2. In the Generate Monitor J2EE Project window (Figure 6-64), for the target project names, accept the default and click **Finish**.

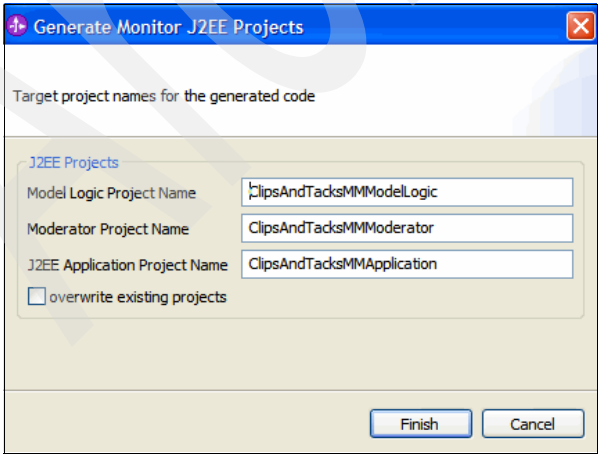


Figure 6-64 Target project names

3. Wait for the project to be generated, and then click **Finish** if prompted. If you refer to the Business Integration perspective now, you see that the Monitor MM J2EE project has been created.
4. In the Servers view, right-click the **WebSphere Business Monitor Server v6.1 on WebSphere ESB** server and select **Add and Remove Projects**.
5. In the Add and Remove Projects window (Figure 6-65), select **ClipsAndTacksMMApplication** and **ClipsAndTacksApp** and click **Add**. Click **Finish** and wait for the application to deploy.

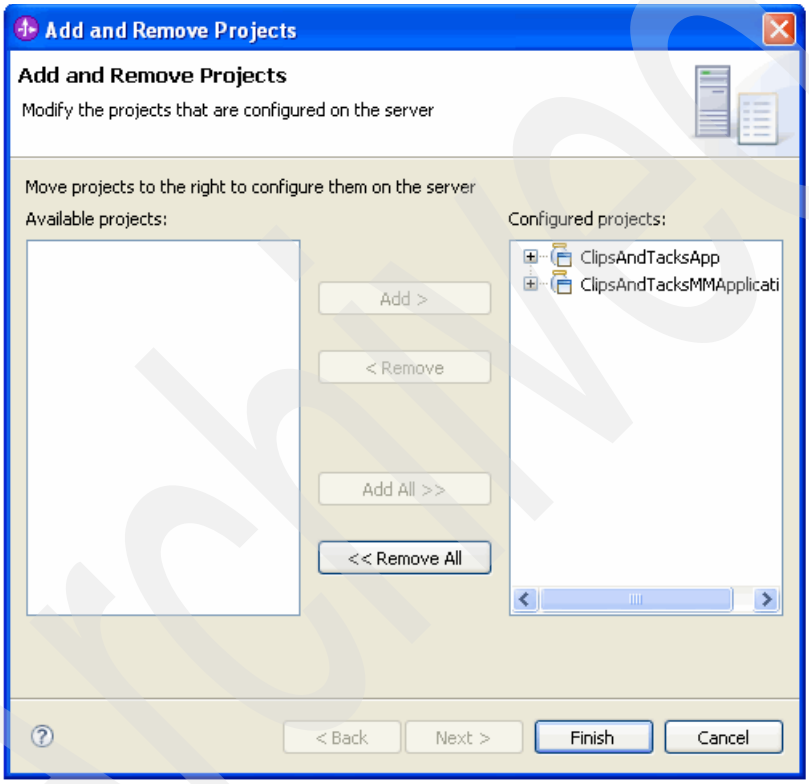


Figure 6-65 Adding ClipsAndTacksMMApplication and ClipsAndTacksApp

Archived

Note: For development purposes, we deployed the monitor model and the monitored application on the WebSphere Business Monitor on WebSphere Enterprise Service Bus test environment in our example.

For production purposes, we recommend the distributed topology. In addition, you should have a stand-alone WebSphere Business Monitor server and a separate WebSphere Enterprise Service Bus server for your monitor model and monitored applications respectively. For more information about this configuration, refer to 3.1.3, “Distributed topology” on page 59.

For detailed instructions about setting up the service integration bus link between the CEI server and WebSphere Business Monitor, refer to “Configuring a remote CEI for WebSphere Business Monitor” on page 218.

6.4.7 Creating the dashboard

We create the dashboard to display the No. of Orders Received KPI that we defined:

1. In the Servers view, right-click **WebSphere Business Monitor v6.1 on WebSphere ESB** and select **WebSphere Business Monitor Dashboard**.
2. If prompted, log in to the dashboard. The default user name and password are `admin`.
3. Click the **Dashboards** tab, and click the **New** button (Figure 6-66).

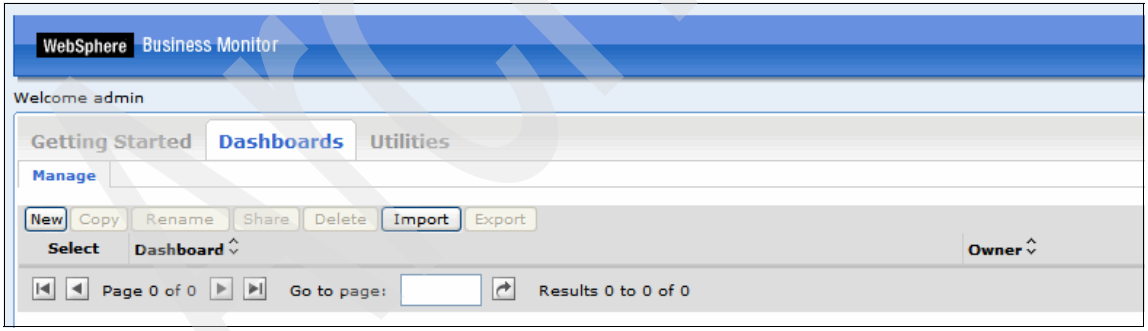


Figure 6-66 Creating a new dashboard

4. Type `OrderDashboard` for the name of the dashboard and click **OK**.

5. In the OrderDashboard view (Figure 6-67), select the **KPI** icon from the palette on the right and drag it to the dashboard panel.

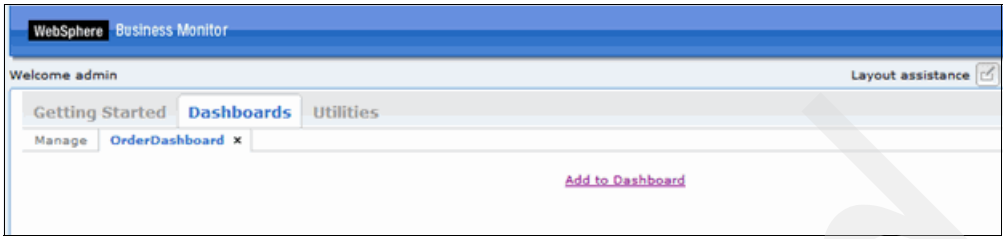


Figure 6-67 Dragging KPI to the dashboard

6. In the KPIs section (Figure 6-68), click the **Personalize** button.

Figure 6-68 Personalizing the KPI

7. Click the **Select KPIs** tab (Figure 6-69). Select the **ClipsAndTacksMM** and **No. of Orders Received** check boxes. Then click **OK**.

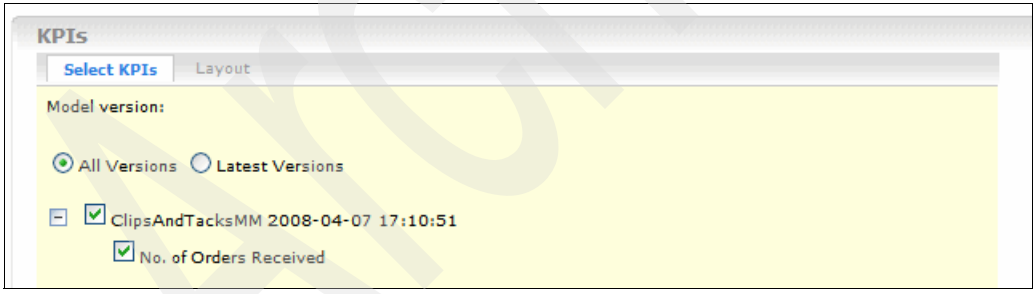


Figure 6-69 Selecting No. of Orders Received

Archived

Figure 6-70 shows the resultant KPI.

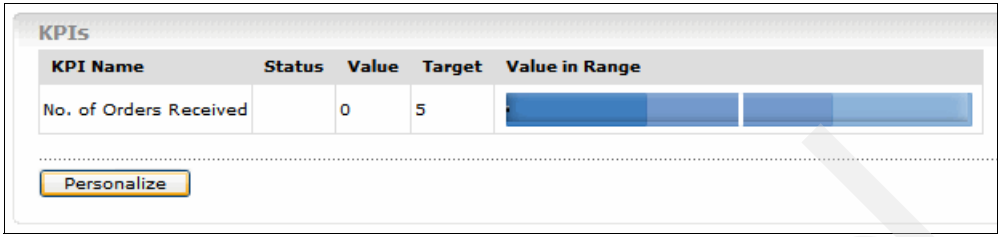


Figure 6-70 No. of Order Received KPI

6.4.8 Testing the service component monitoring scenario

Test the first scenario, which involves service component monitoring. Ensure that the order records are deleted. In the DB2 Command Line Processor, run the **`LoadOrder.sql`** script (Example 6-3 on page 290).

If you are successful, you see the results updated on the dashboard (Figure 6-71) after some time.

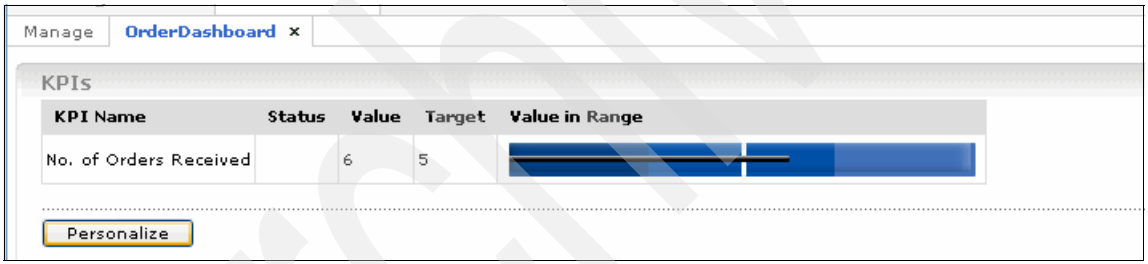


Figure 6-71 KPI results

You have completed the example for employing adapters to retrieve events from an EIS and seen how we can monitor the generic service component events in WebSphere Business Monitor. Before you continue on with the next scenario, you must clean up the dashboard as explained in the following section.

Archived

Cleaning up the dashboard

After you complete this section, remove the No. of Order Received KPI from the dashboard:

1. Click the menu icon in the upper right corner of the KPI portlet and select **Delete** (Figure 6-72).

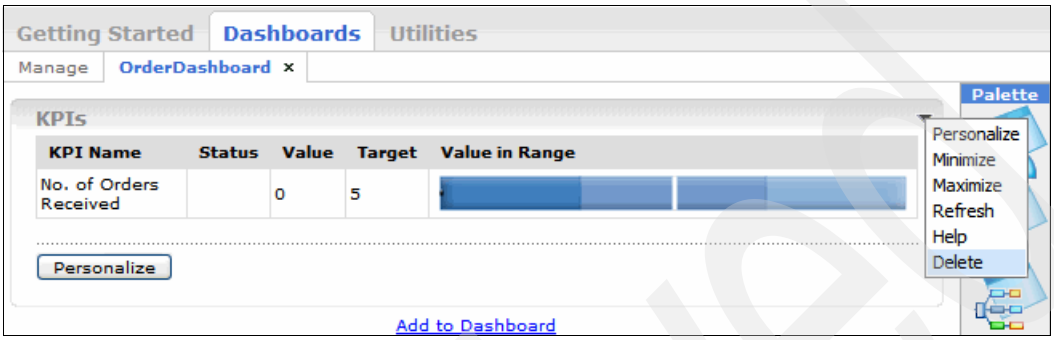


Figure 6-72 Removing the KPI

2. Remove all the projects from the server before moving to the next section. In the Server view, right-click the **WebSphere Business Monitor Server v6.1 on WebSphere ESB** server and select **Add and Remove Projects**.
3. In the Add and Remove Projects view, click **Remove All**. Click **Finish**.
4. Connect to the ORDER database again and delete the records from the tables (Example 6-4 on page 291).

6.5 Scenario 2: Monitoring mediation flow custom events

In this section, we discuss the use of the Event Emitter primitive that is available in the mediation flow component for generating custom events. We use the same fictitious ClipsAndTacks Office Supplies scenario (Figure 6-73 on page 312).

Archived

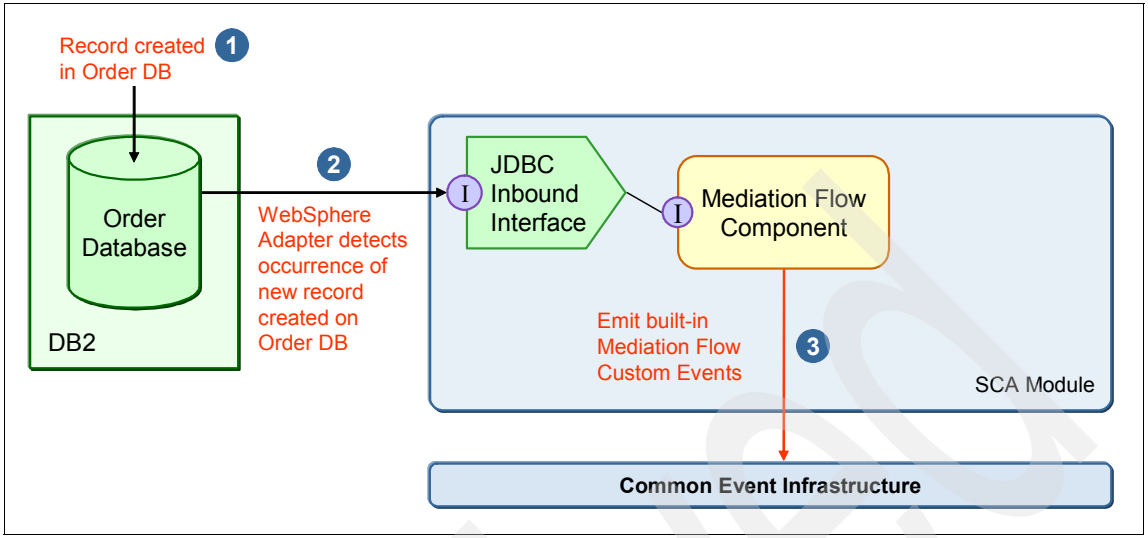


Figure 6-73 Scenario 2 flow

For a discussion about why you might use the Event Emitter primitive, refer to 6.7.2, “When to use built-in service component events and Event Emitter primitive” on page 348.

6.5.1 Creating the mediation module

In this section, we explain how to create the mediation flow component and a mediation flow in the Mediation Flow editor.

Creating the mediation flow component

A mediation flow is a flow-like construct created with the Mediation Flow editor. In the Mediation Flow editor, an operation from one service, the service requester or export, is mapped to the operation of another service, the service provider or import, along with functions provided by the Mediation Flow editor. These functions are called *mediation primitives* and are wired in a mediation flow. In this scenario, we examine the use of the Event Emitter primitive in a mediation flow.

To create the mediation flow component:

1. Go back to the assembly diagram. Click the **Mediation Flow** icon on the palette and add it to the canvas. Rename the mediation flow component `OrderMediation` (Figure 6-74).

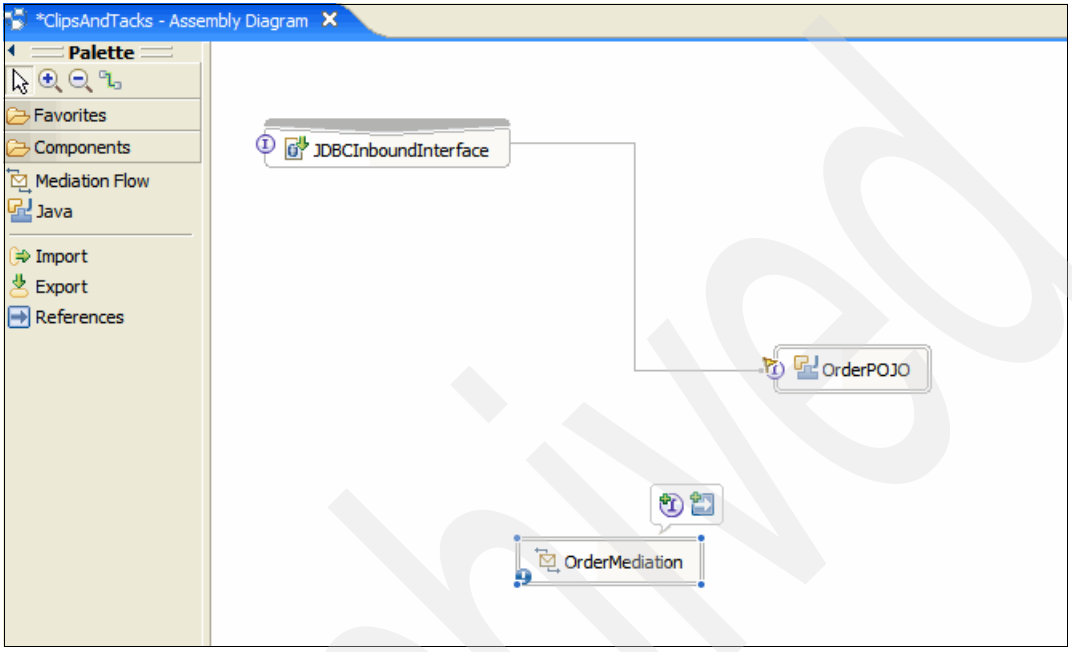


Figure 6-74 Adding the OrderMediation mediation flow

2. Select the link between **JDBCInboundInterface** and **OrderPOJO** and delete it (Figure 6-75).

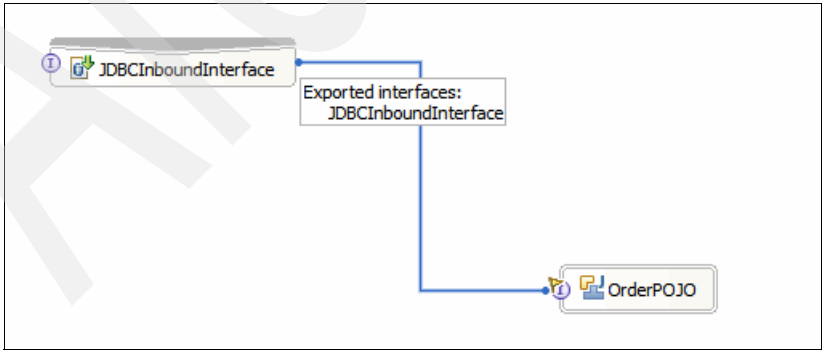


Figure 6-75 Deleting the link between JDBCIboundInterface and OrderPOJO

Archived

3. Wire **JDBCInboundInterface** to **OrderMediation**. If prompted, click **OK** to add the service interface from the export to the target (Figure 6-76).

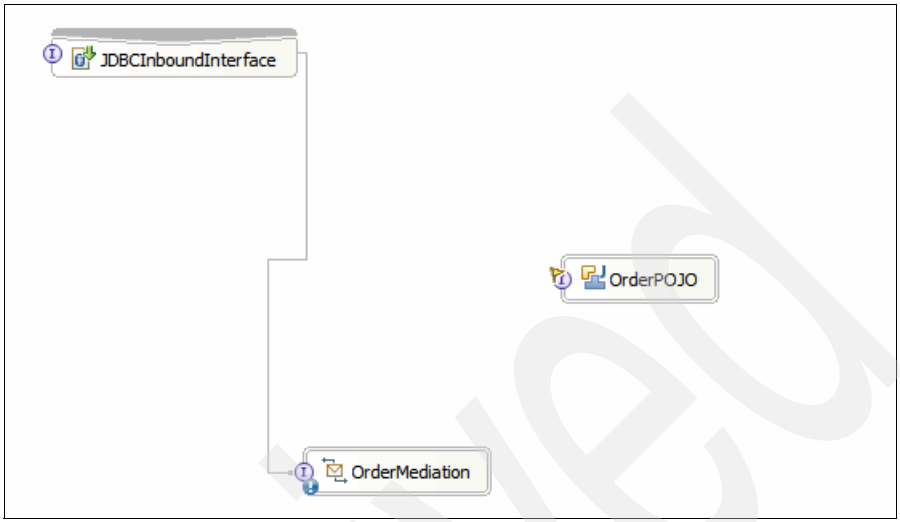


Figure 6-76 Rewiring to OrderMediation

4. Right-click **OrderMediation** and select **Generate Implementation**. When prompted, select the default location for the implementation (Figure 6-77).

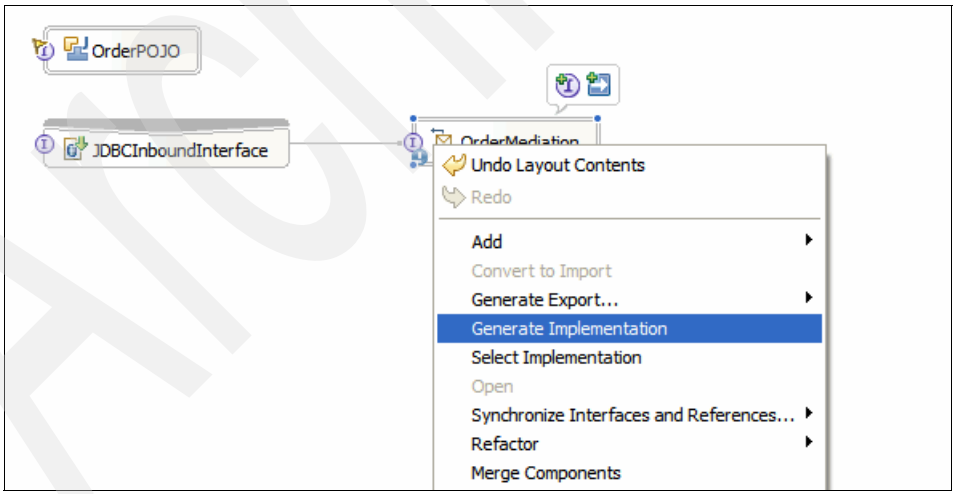


Figure 6-77 Generating the implementation of the mediation flow

Archived

Creating a mediation flow in the Mediation Flow editor

The Mediation Flow editor is a graphical programming environment that you use to visually create and manipulate mediation flows. You can visually compose a mediation flow by defining the source and target operations in the Operation connections section of the Mediation Flow editor, and then visually adding and wiring mediation primitives in the flow section.

In this section, we create the mediation flow for filtering the data that arrives in the WebSphere Enterprise Service Bus. We only emit events from the mediation flow if the orders received are larger orders, that is more than USD $20.

Message Filter primitive

To create a mediation flow for the Message Filter primitive:

1. From the Palette, select **Routing** → **Message Filter** and add it to the canvas. Wire the **Input node** to the **Message Filter** primitive (Figure 6-78).

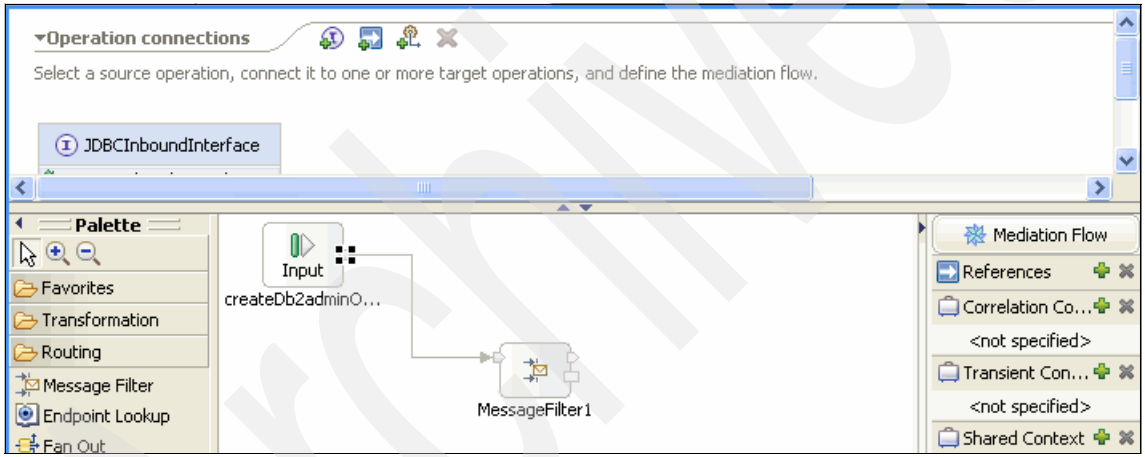


Figure 6-78 Wiring the input node to a message filter primitive

Archived

2. Right-click the **Message Filter** primitive and select **Add Output Terminal** (Figure 6-79). An output terminal and an error terminal are created by default. The bulk of the inputs received by the Message Filter is routed to the default output terminal. We must create an additional output terminal for the rest of the input that meets the conditions specified in the Message Filter primitive.

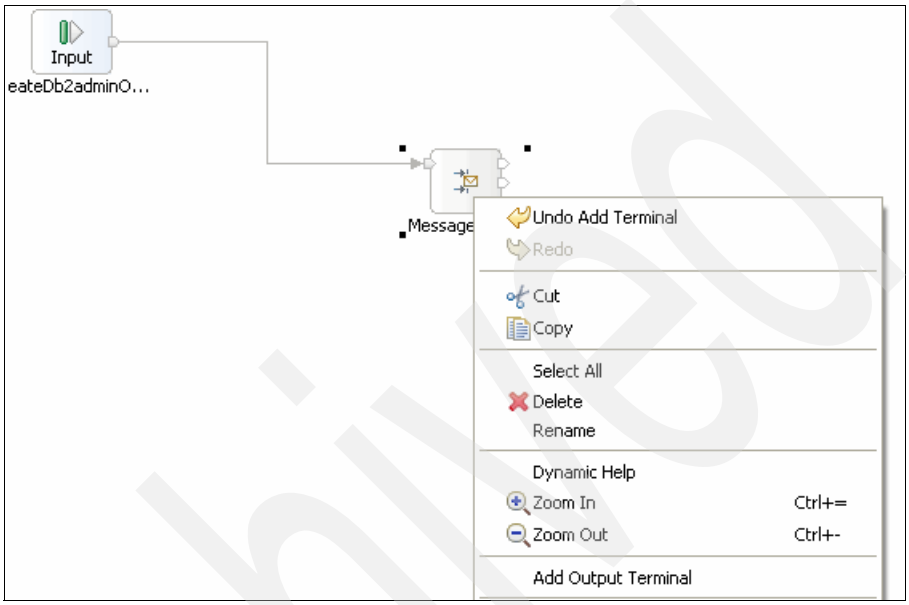


Figure 6-79 Selecting Add Output Terminal

3. In the New Dynamic Terminal window (Figure 6-80), click **OK** to create a new output terminal.

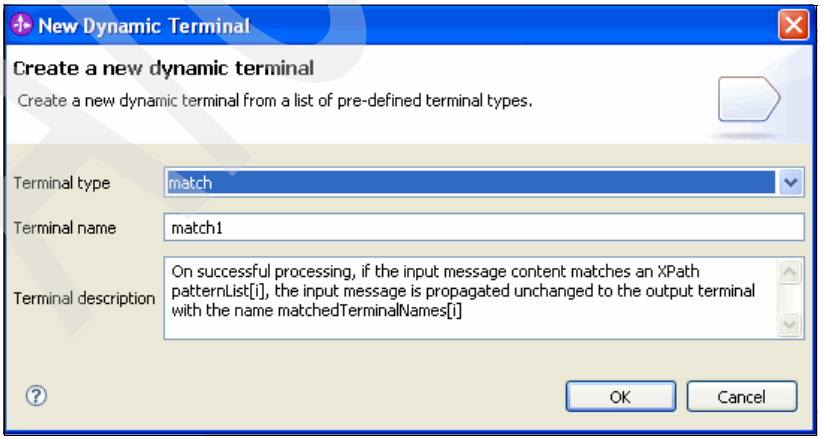


Figure 6-80 Creating a new output terminal

Archived

4. Select the **Message Filter** primitive. Then click the **Properties** tab and the **Details** subtab (Figure 6-81) and click **Add**.

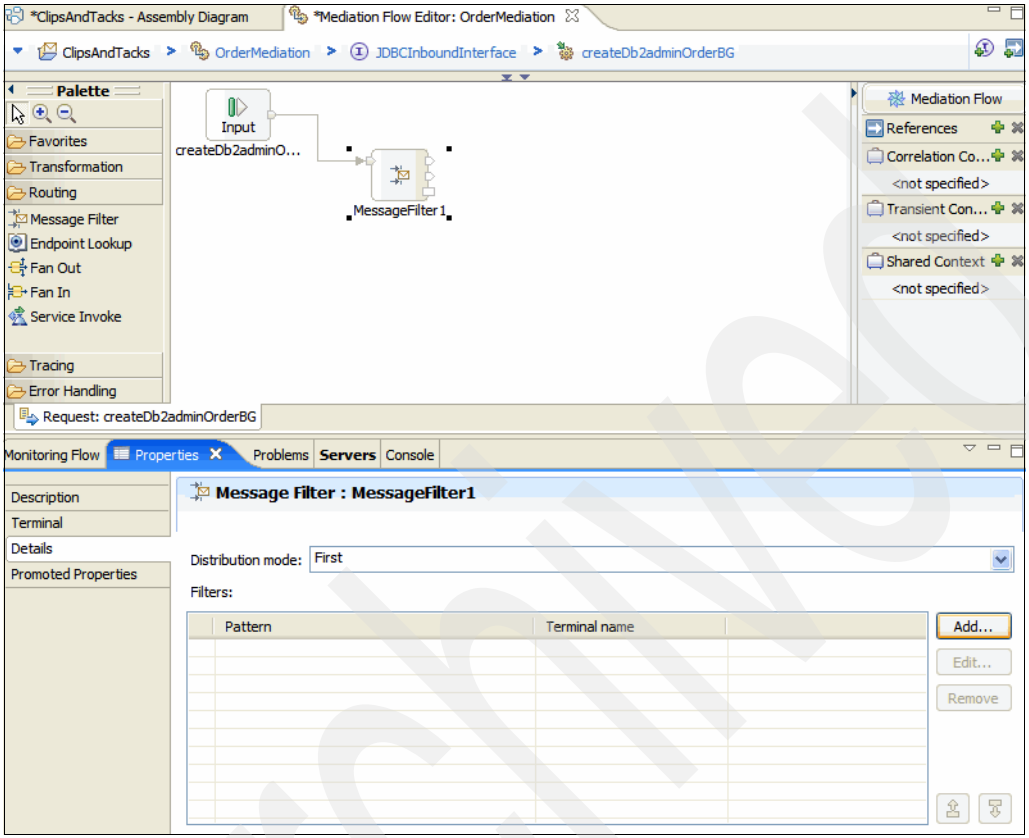


Figure 6-81 Adding filters

Archive

5. In the Add/Edit window (Figure 6-82), click **Edit**.

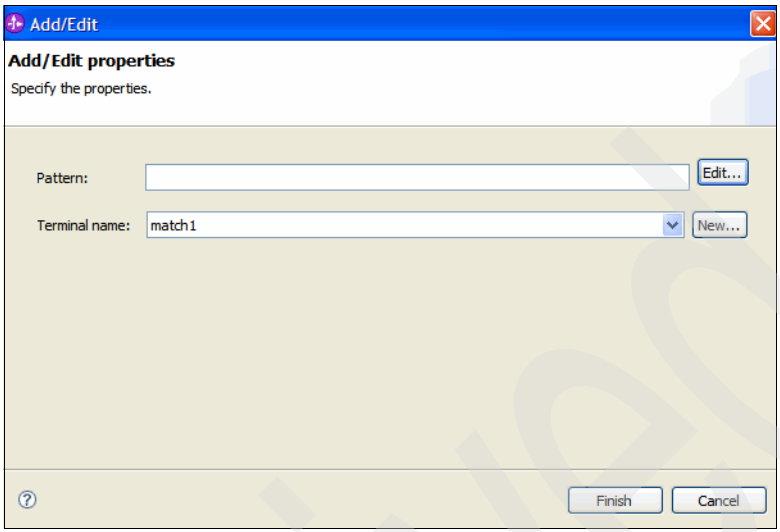


Figure 6-82 Add/Edit window

Archived

6. In the XPath Expression Builder window (Figure 6-83), create a filter for filtering out large orders:
   a. Under Data Types Viewer, expand **ServiceMessageObject** → **body** → **createDb2adminOrderBG** → **createDB2adminOrderBGInput** → **Db2adminOrder** and select **tprice**.
   b. Under Operators, double-click **>**.
   c. For XPath Expression, type `/body/createDb2adminOrderBG/createDb2adminOrderBGInput/Db2adminOrder/tprice > 20`.
   d. Click **Finish**.

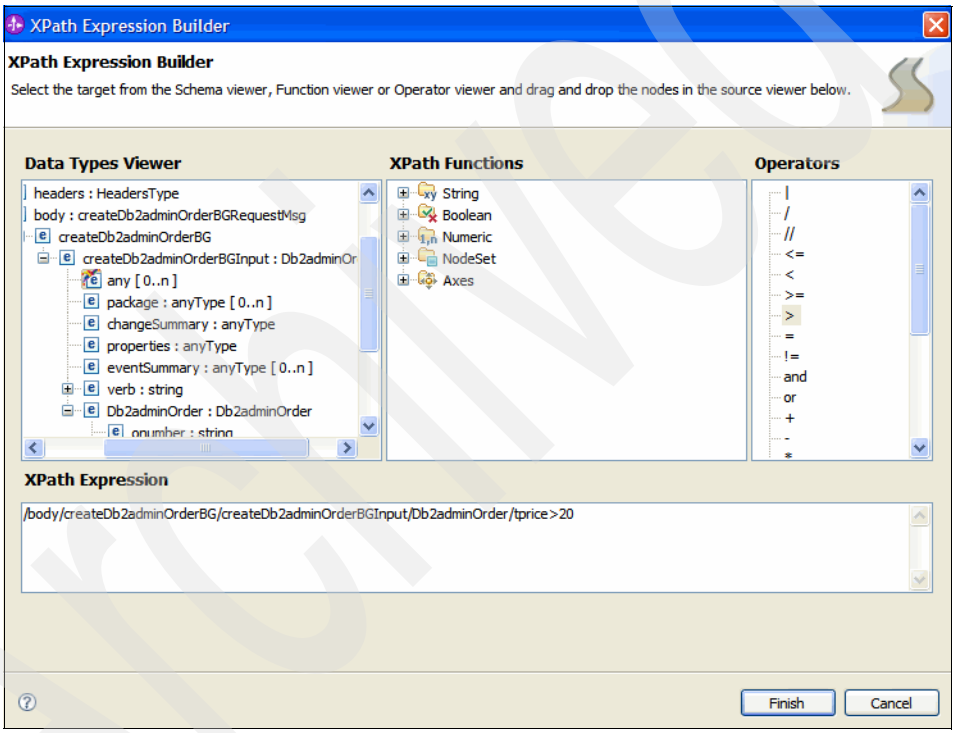


Figure 6-83 XPath Expression Builder

7. Back in the Add/Edit window, click **Finish**.
8. In the Mediation Flow editor, add two **Custom Mediation** primitives from the Palette. The Custom Mediation primitives are in the Transformation section.

Archived

9. Under the Tracing section, select and add an **Event Emitter** primitive to the canvas (Figure 6-84).

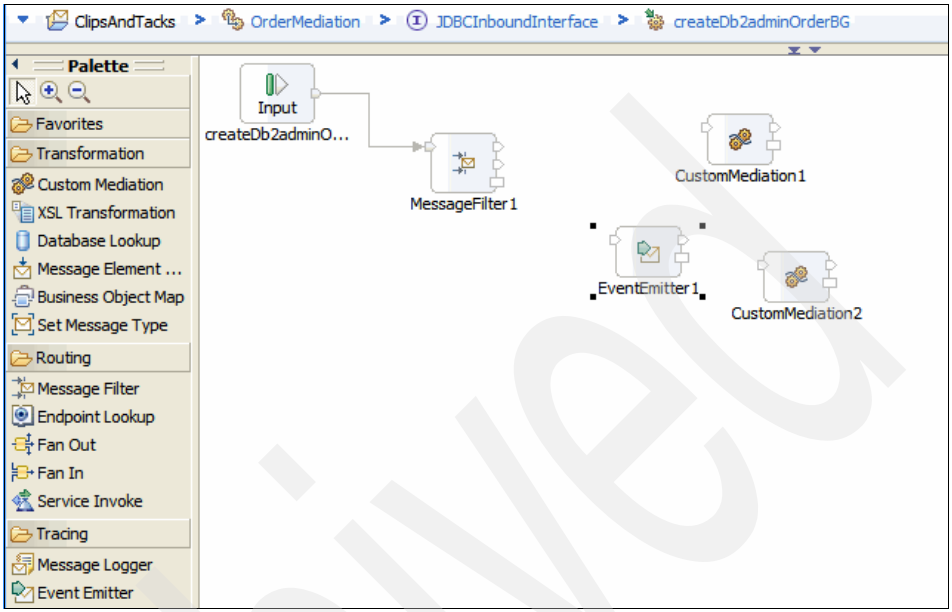


Figure 6-84 Adding primitives

Custom mediation primitive

By using the custom mediation primitive, you can implement your own mediation logic in Java code. The primitive that we create in the following steps prints the type of orders that is receives. It is used to verify that the message filter is functioning as expected.

1. Rename the custom mediation primitives to `Normal Order` and `Large Order` and connect the primitives together as in Figure 6-85. Wire Normal Order to the default terminal of Message Filter and Event Emitter to the match1 terminal of Message Filter. In this way, orders that are received by the Message Filter are routed to Normal Order by default. Large orders are routed to Large Orders, and an event is emitted by the Event Emitter primitive.

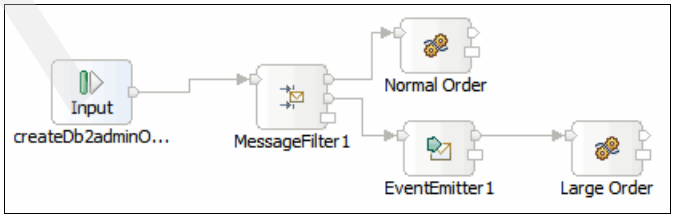


Figure 6-85 Connected primitives

2. Select the **Normal Order** custom mediation primitive. Then click the **Properties** tab and the **Details** subtab (Figure 6-86). For Implementation, select **Visual**. When prompted to continue, click **Yes**.

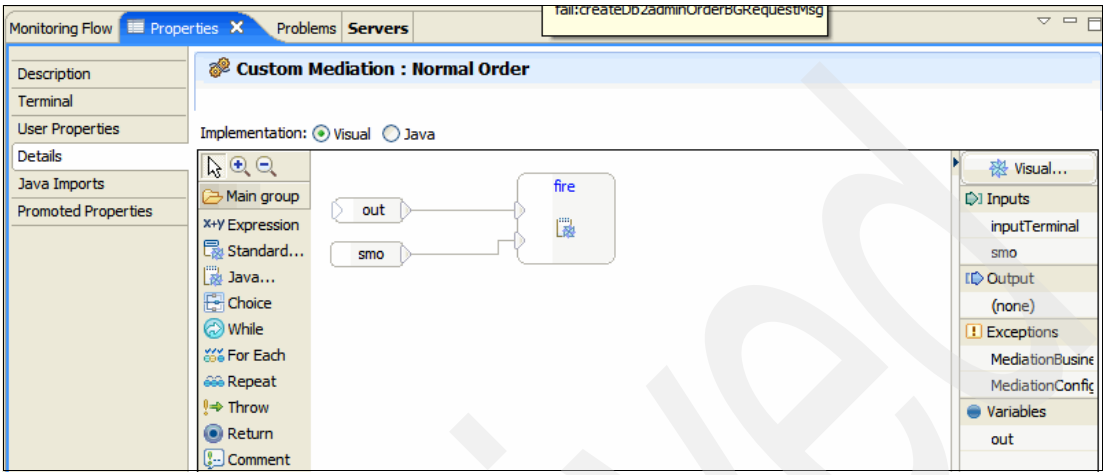


Figure 6-86 Visual Snippet Editor

3. From the Palette, select **Standard visual snippet**.
4. Select **utility** → **print to log** and click **OK**. Add it to the canvas.
5. Select the **expression snippet** and add it to the canvas as well.
6. Connect the **expression** and **standard visual snippets** (Figure 6-87).

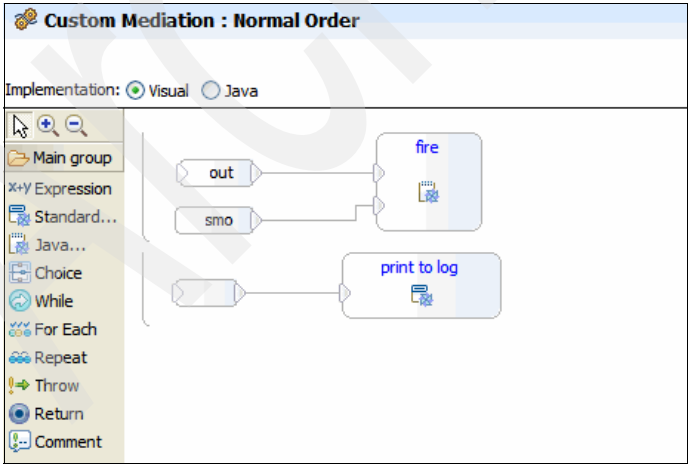


Figure 6-87 Adding the visual snippets

Archived

7. Click the **expression snippet** and type "`Normal Order`" (Figure 6-88).

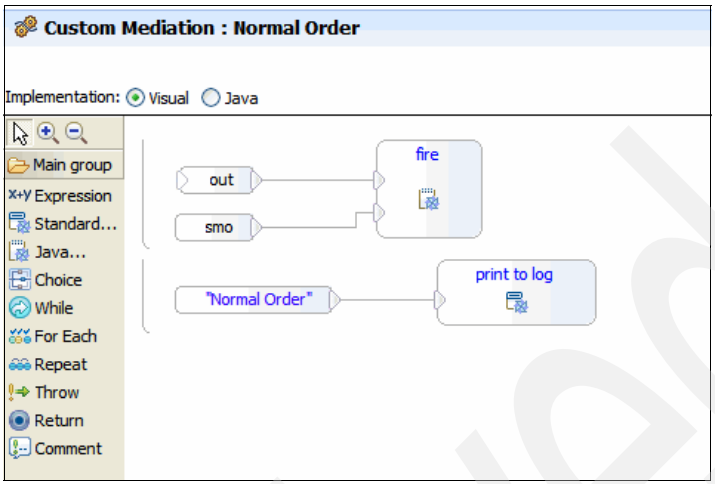


Figure 6-88 Normal Order

8. Repeat step 2 on page 321 through step 7 to modify the Large Order custom mediation to output "`Large Order`" to the log (Figure 6-89).

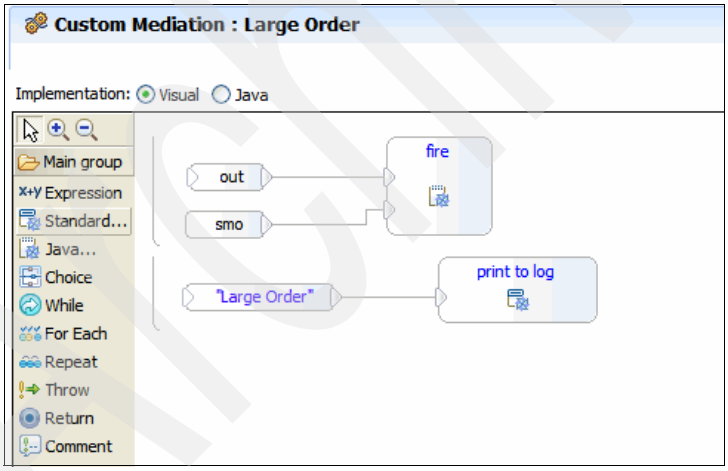


Figure 6-89 Large Order

Archived

Event Emitter primitive

The Event Emitter primitive emits Common Base Events only on the occurrence of a particular situation. Common Base Events are emitted in our scenario only if the orders are large.

1. In the Mediation Flow editor, select the **Event Emitter** primitive.
2. Click the **Properties** tab and the **Details** subtab (Figure 6-90). Click **Edit**.

> **Custom events:** In this section, you can define the custom events to be emitted by the Event Emitter primitive.

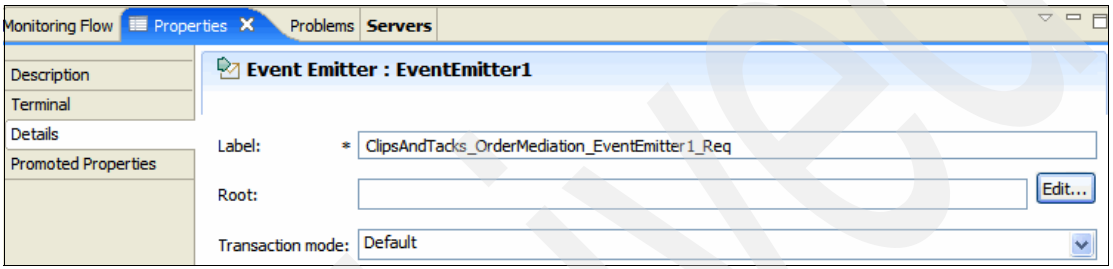


Figure 6-90 Event Emitter properties

Archived

3. Define the business object that is emitted by the Event Emitter primitive. In the XPath Expression Builder window (Figure 6-91), expand **ServiceMessageObject** → **body**. Click **Finish**.

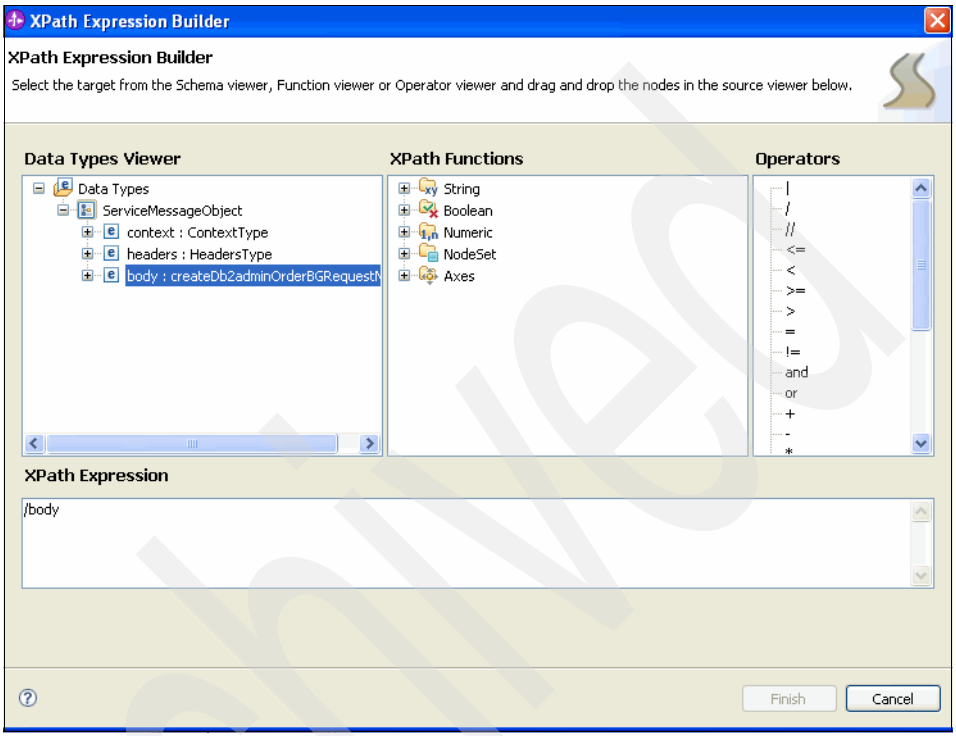


Figure 6-91 XPath Expression Builder window

Stop mediation primitive

The Stop mediation primitive is used to stop a path in the flow, without generating an exception because there are no further actions:

1. To complete the mediation flow, add two **Stop** mediation primitives under the Error Handling section and connect the primitives to the mediation flow. (Figure 6-92 on page 325).

Archived

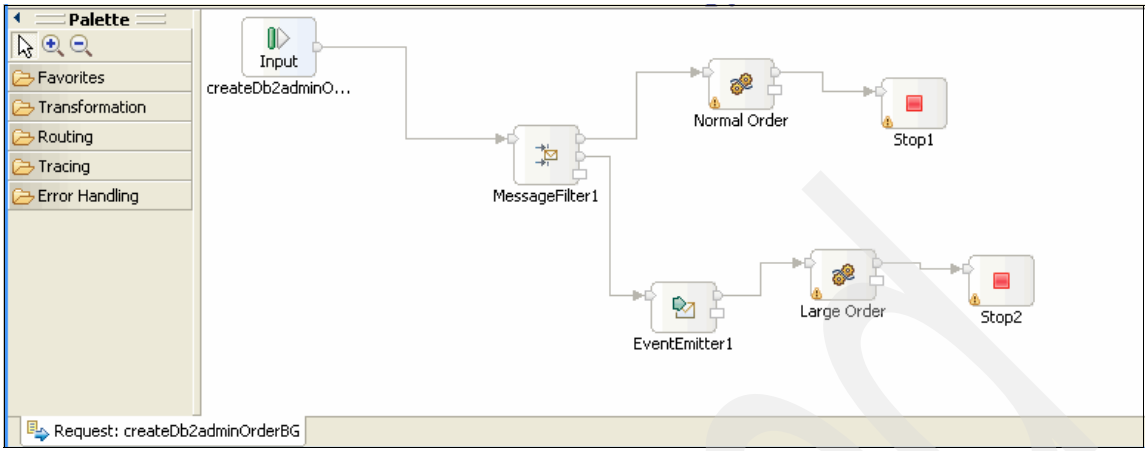


Figure 6-92 Adding Stop mediation primitives

2. Save your work.
3. To remove the warnings, select the **Normal Order** custom mediation primitive (Figure 6-93). In the Properties view, click the **Terminal** subtab and select **Output terminal → out**.

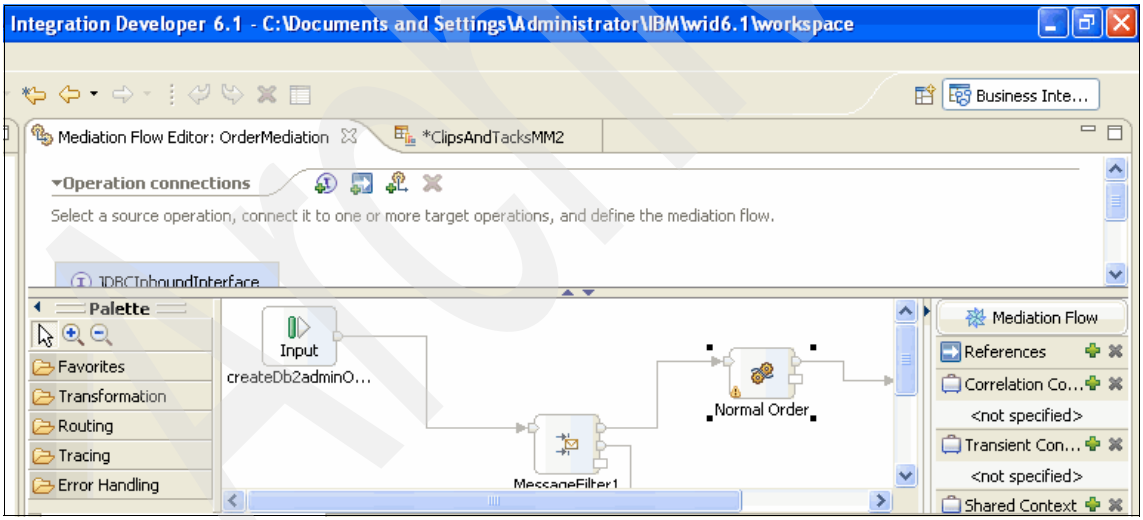


Figure 6-93 Output Terminal message type

4. In the Message type field, click **Change**.

5. In the Change Message Type window (Figure 6-94), click **Browse**, and for Interface, select **JDBCInboundInterface**.

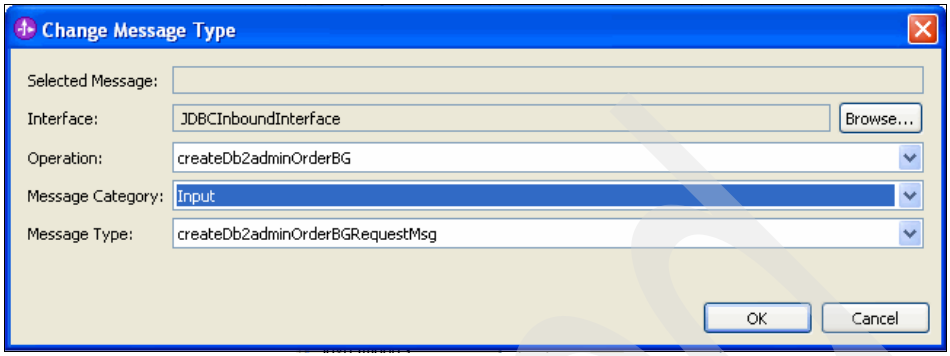


Figure 6-94 Change message type

6. For Operation, select **createDb2adminOrderBG**, for Message Category, select **Input**, and for Message type, select **createDb2adminOrderBGRequestMsg**.
7. Select the Large Order custom mediation primitive and repeat step 3 on page 325 through step 6 outlined for the Normal Order custom mediation primitive.
8. Save your work. There are no warnings now (Figure 6-95).

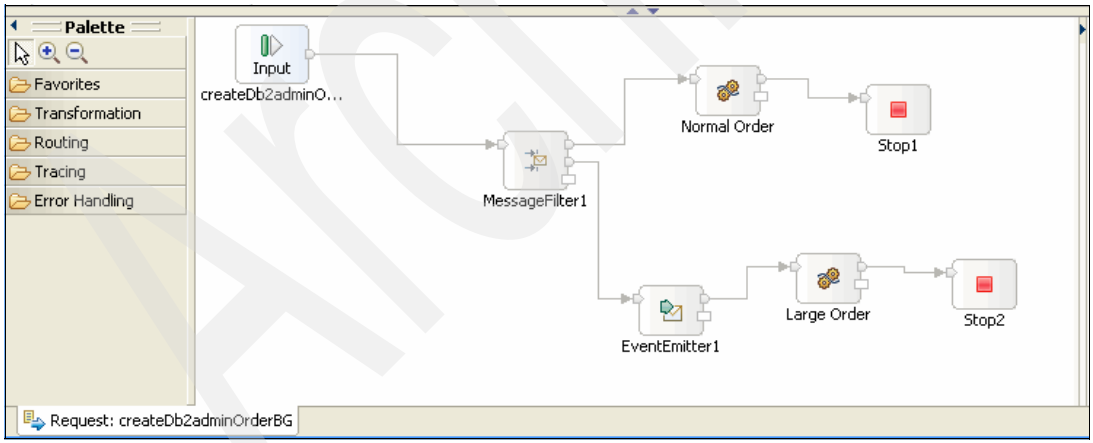


Figure 6-95 No warnings

6.5.2 Enabling event generation

To enable event generation:

1. In the Assembly Diagram editor:
   a. Select the **OrderMediation** component.
   b. Click the **Properties** tab and the **Details** subtab (Figure 6-96).
   c. Expand **Interfaces** → **JDBCInboundInterface** → **createDb2adminOrderBG**
   d. Click the **Event Monitor** tab. For Monitor, select **All** and for Event Content, select **Full**. This indicates that the option is selected for emitting all the events for the createDb2adminOrderBG operation. The full content of the event, instead of a digest, is also sent.

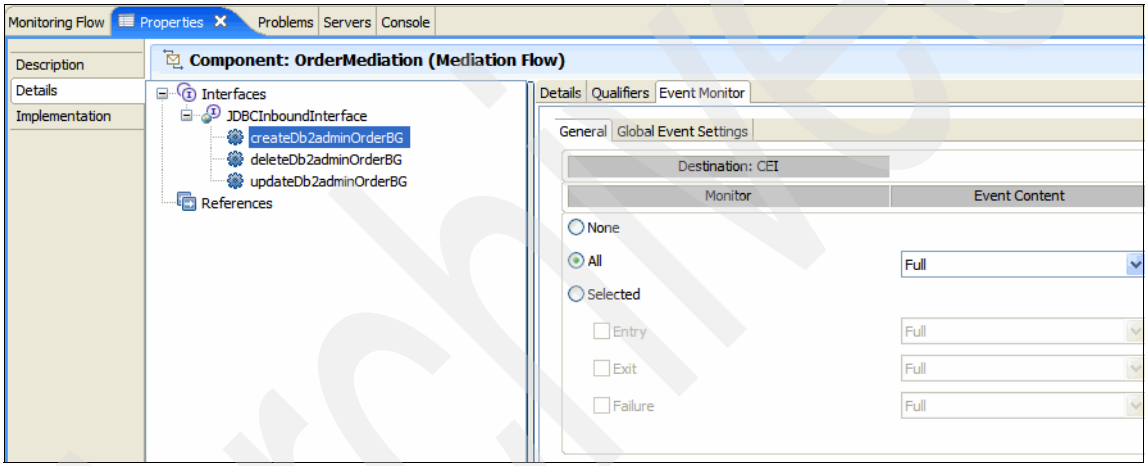


Figure 6-96 General Event Settings of OrderMediation component

2. Save your work. Close the assembly diagram.
3. Rebuild the project from the beginning to resolve any broken links before deploying your project. In the menu bar, select **Project** → **Clean**. Select **Clean All Projects** and click **OK**.

Creating the monitor model

In this section, we explain how to augment the existing monitor model and generate the new monitor J2EE project.

Archived

Augmenting the existing monitor model

To augment the existing monitor model:

1. In the Business Integration perspective, right-click the **ClipsAndTacks** mediation module and select **Monitor Tools** → **Generate Monitor Model**.
2. In the Generate Monitor Model window (Figure 6-97), for Target monitor project, select **ClipsAndTacksBM**, and for Target monitor model name, type `ClipsAndTacksMM2`. Then click **Next**.

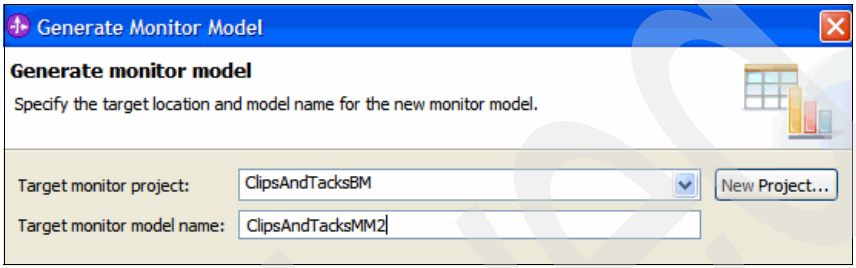


Figure 6-97 Specifying the monitor model location

3. In the Choose what to monitor window (Figure 6-98):
   a. Select the option to monitor the event generated by the Event Emitter primitive:
      i. Expand **OrderMediation** → **EventEmitter1**.
      ii. Click the **Emitted Events** tab.
      iii. Select the **ClipsAndTacks_OrderMediation_EventEmitter1_ReqCUSTOM** event.

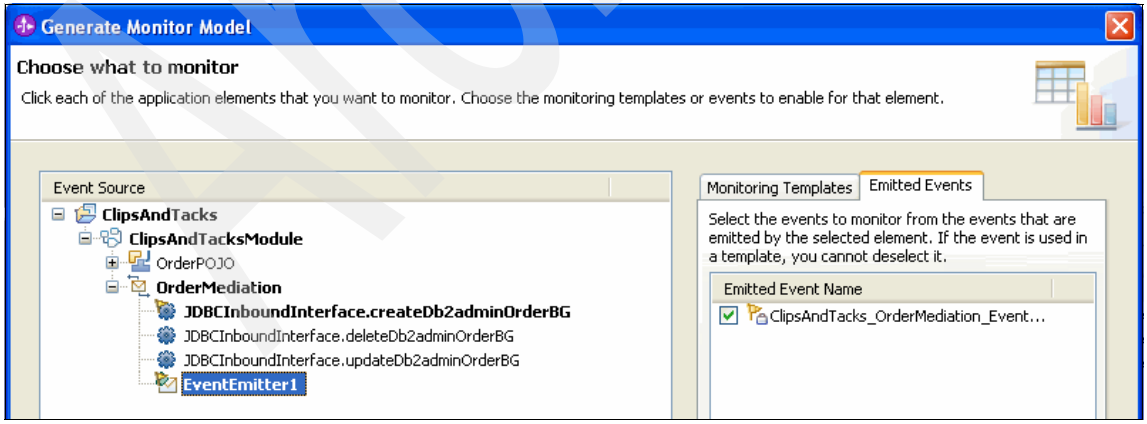


Figure 6-98 Events emitted from the Event Emitter primitive

b. Select the option to monitor all the events emitted when the createDb2adminOrderBG operation is invoked (Figure 6-99)

   i. Select **JDBCInboundInterface.createDb2adminOrderBG**.

   ii. Click the **Emitted Events** tab and select all the events listed.

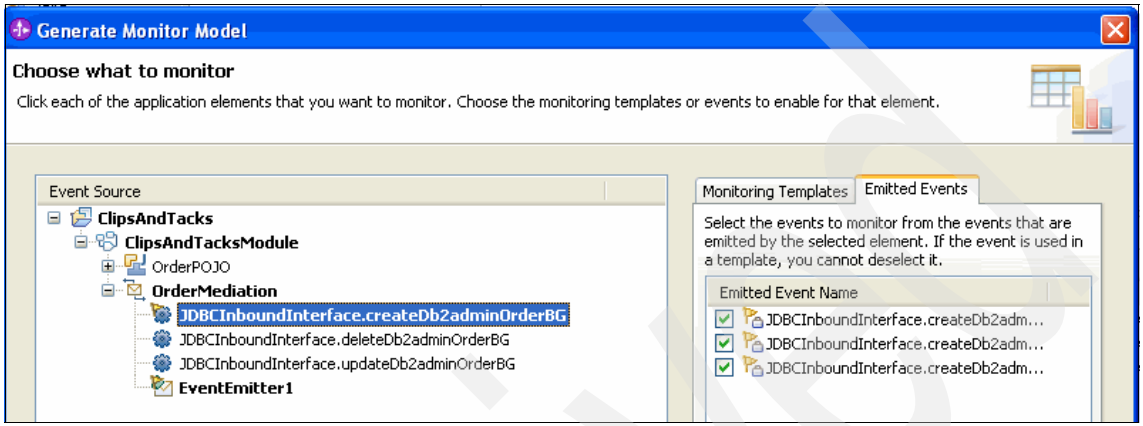


Figure 6-99 Events emitted from createDb2adminOrderBG operation

c. Click **Next**.

4. In the Choose how to monitor window, click **Next**.
5. In the Preview the monitor model window, click **Expand All** and review that you have configured the events correctly. Figure 6-100 shows the results. Click **Finish**.

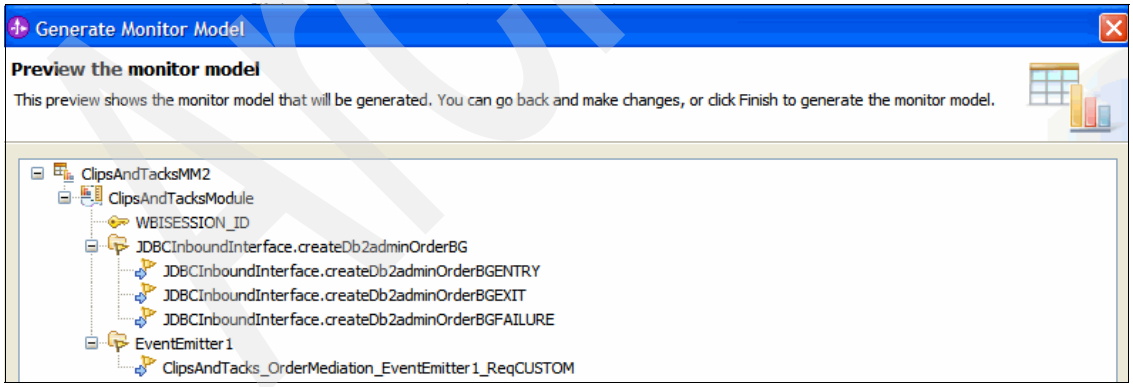


Figure 6-100 Previewing the monitor model

6. When prompted to switch to the Business Monitoring perspective, click **Yes**.
7. When prompted to launch the Get Started information, click **No**.

8. Ensure that there are no warning signs on ClipsAndTacksMM2.mm. If there are, review the steps in this section again before proceeding:
   a. Open the assembly diagram. Ensure that the event generation is enabled for the createDb2adminOrder operation.
   b. Delete **ClipsAndTacksMM2.mm**.
   c. Regenerate the monitor model again.

Merging with existing monitor model

You can compare two monitor models to see their differences and merge them. This capability is useful when one of the monitor models started as a copy of another, and therefore, they have some ID similarities.

1. In the Project Explorer, right-click **ClipsAndTacksMM.mm** and select **Synchronize with Application**.
2. In the Synchronize Monitor Model window (Figure 6-101), click **OK**.

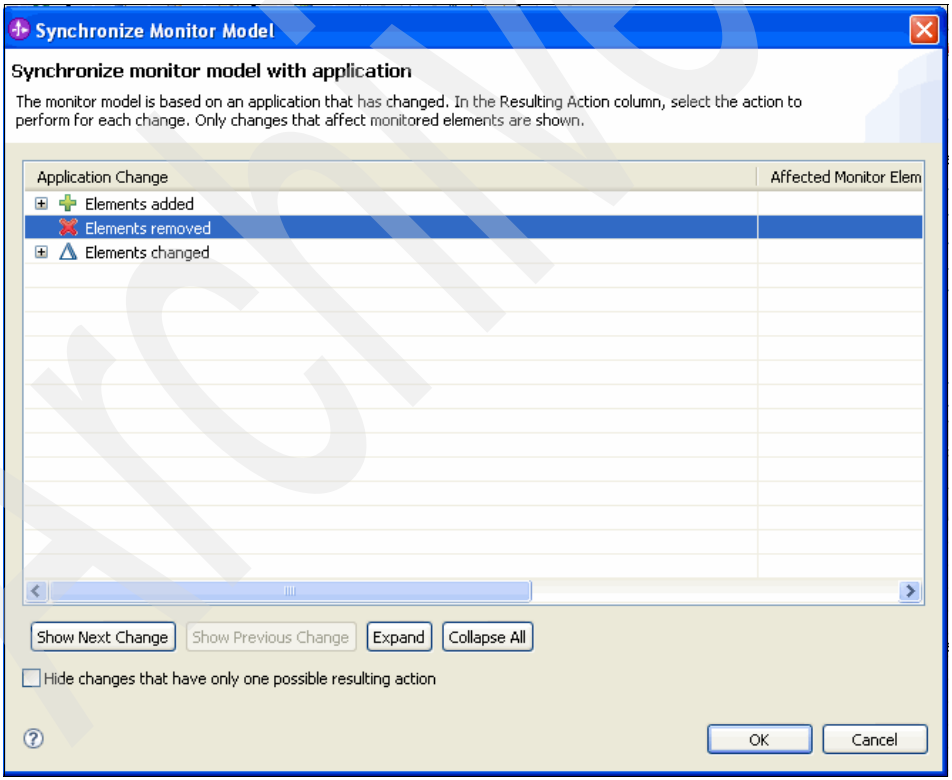


Figure 6-101 Synchronize Monitor Model window

Archived

3. In the Project Explorer, select **ClipsAndTacksMM.mm**. Press Ctrl and select **ClipsAndTacksMM2.mm**. Right-click **Compare With** and select **Each Other**.
4. The two monitor model files open in a monitor model comparison editor (Figure 6-102), with the changes highlighted. The Structural Changes pane shows the differences between the two monitor models. The changes are described in terms of the actions that must be performed on the new monitor model to make it identical to the original monitor model.
   a. Select the **Add Order_Received to ClipsAndTacksModule** metric and click the **Accept** icon.
   b. Select **Add KM to ClipsAndTacksMM2 kpiModel** and click the **Accept** icon.

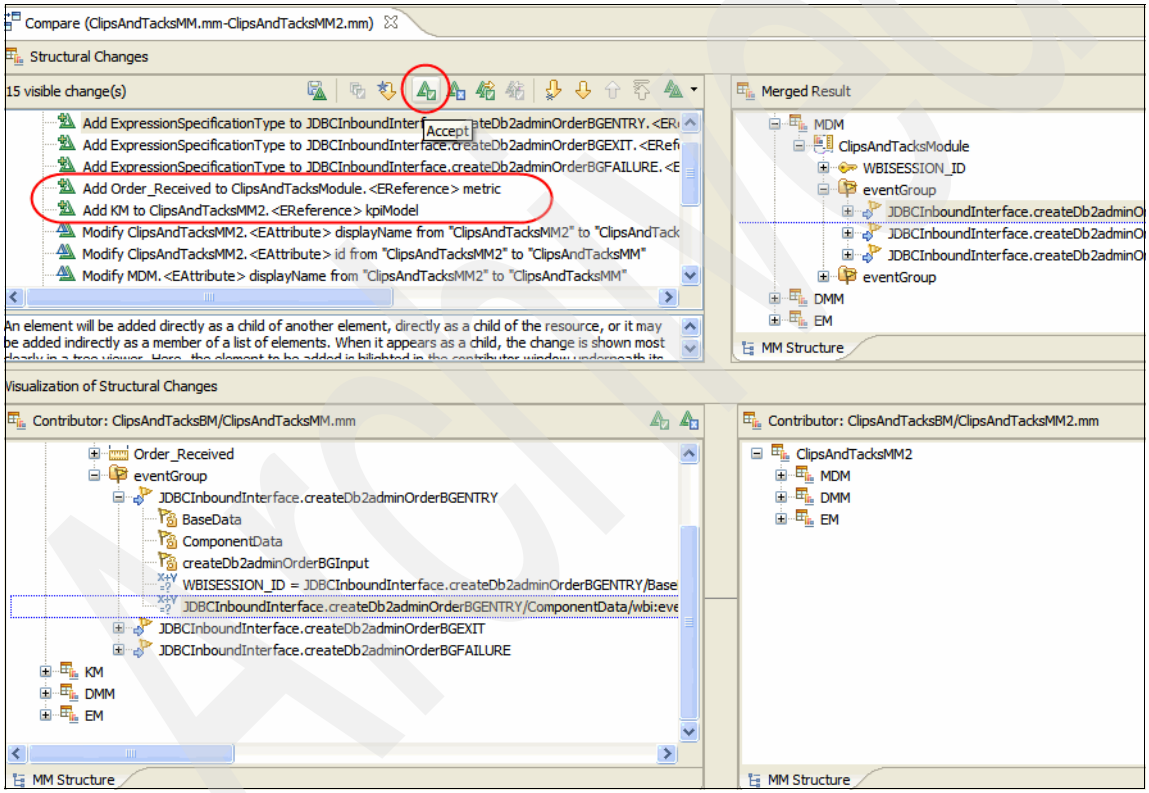


Figure 6-102 Compare monitor model editor

Note: In our scenario, the two monitor models were generated from the same mediation module. The difference is that we defined the metrics and KPI in the first model and defined an Event Emitter primitive.

We could have taken two alternative approaches: modifying the original monitor model to include the new inbound event from the event emitter or recreating the KPI and business metric in the new monitor model. We chose to use the merge capability, so that we can achieve the final monitor model quickly.

5. In the Merged Result panel (Figure 6-103), click the **Save as Right Contributor** icon.

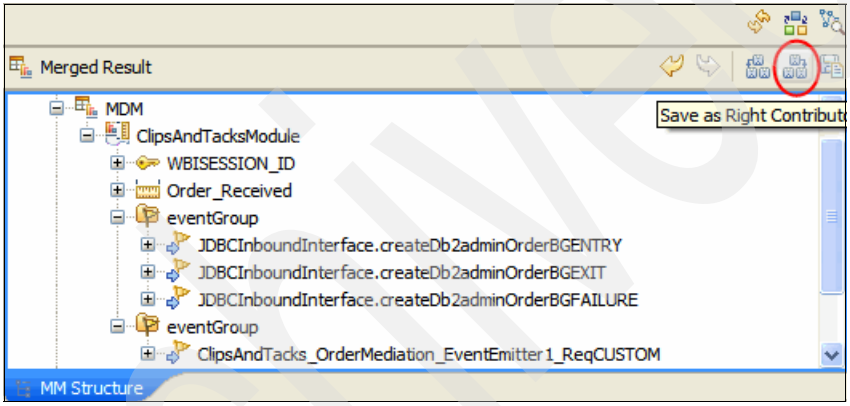


Figure 6-103 Clicking the Save as Right Contributor icon

6. When prompted by the Commit Merge Session window (Figure 6-104) to ignore the unresolved differences and complete the merge, click **Yes**.

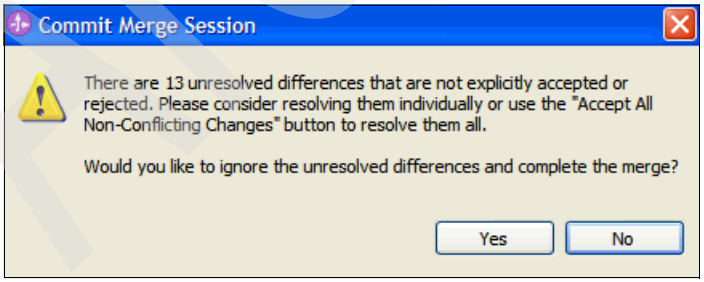


Figure 6-104 Commit Merge Session

7. Close the compare view.

Adding a new business metric

To add a new business metrics for large orders:

1. In the monitor model editor (Figure 6-105), expand **ClipAndTacksMM2**, right-click **ClipsAndTacksModule**, and select **New** → **Metric**.

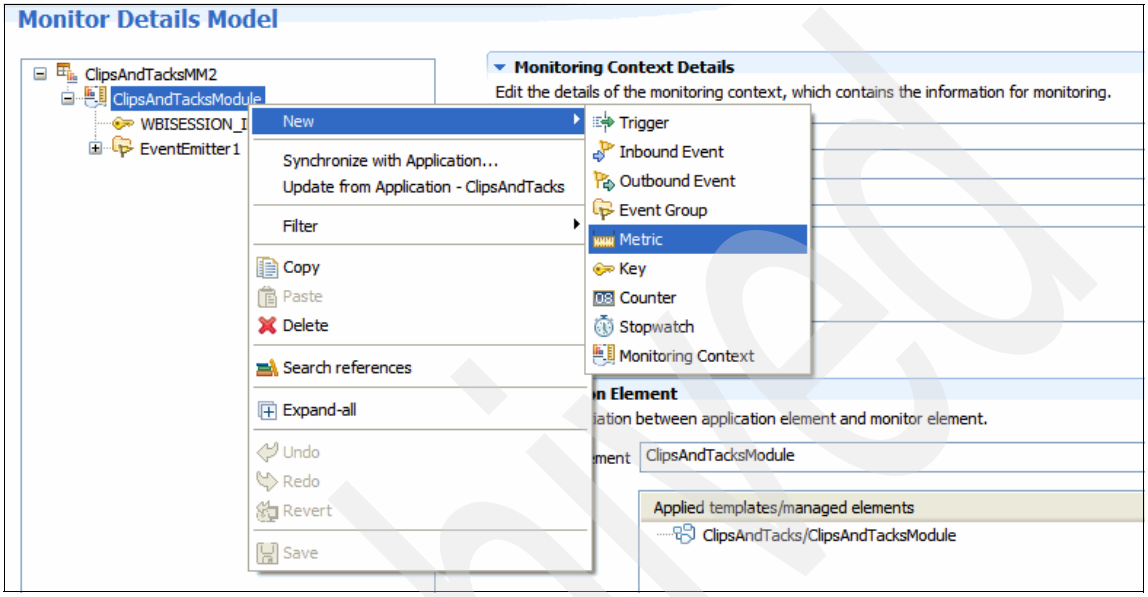


Figure 6-105 Creating a new metric

2. For the name of the new metric, type `Large Order` and click **OK**.

Archived

3. In the Monitor Details Model panel (Figure 6-106), select the newly created **Large Order** metric. Under Metric Value Expressions, on the right, click **Add**. Select the newly added expression and click the **...** button.

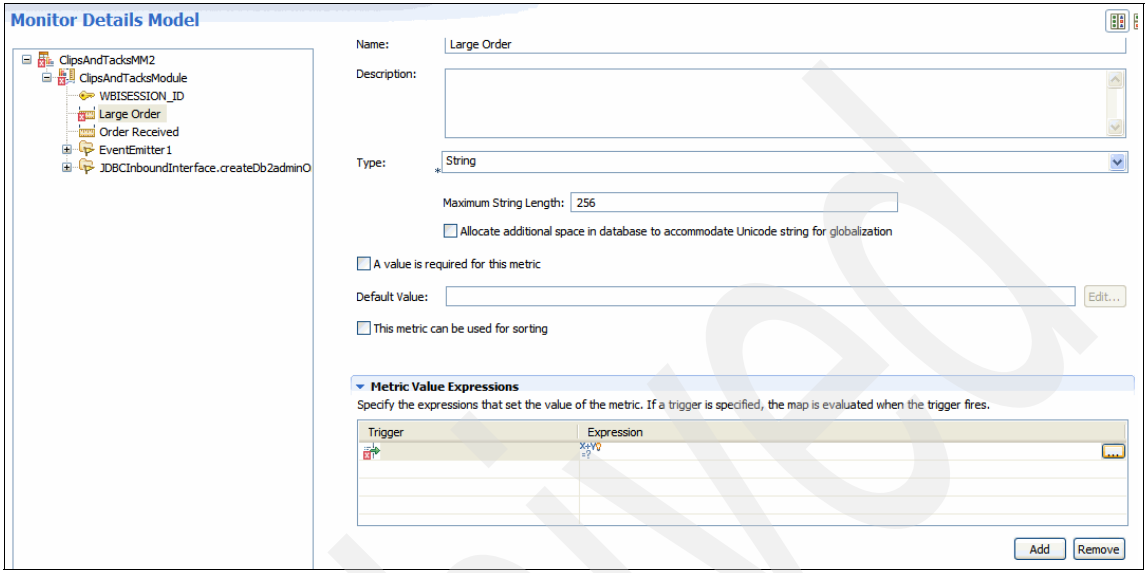


Figure 6-106 Adding a Metric Value Expression

4. In the Expression Dialog, press Ctrl+Spacebar.

Archived

5. As shown in Figure 6-107, expand **ClipsAndTacksMM2** → **ClipsAndTacksModule** → **ClipsAndTacks_OrderMediation_EventEmitter1_ReqCUSTOM** → **Message** → **createDb2adminOrderBG** → **createDb2adminOrderBGInput** → **Db2adminOrder** and select **tprice**. Click **OK**.

   The completed expression is `ClipsAndTacks_OrderMediation_EventEmitter1_ReqCUSTOM/Message/tns1:createDb2adminOrderBG/tns1:createDb2adminOrderBGInput/Db2adminOrder/tprice`.

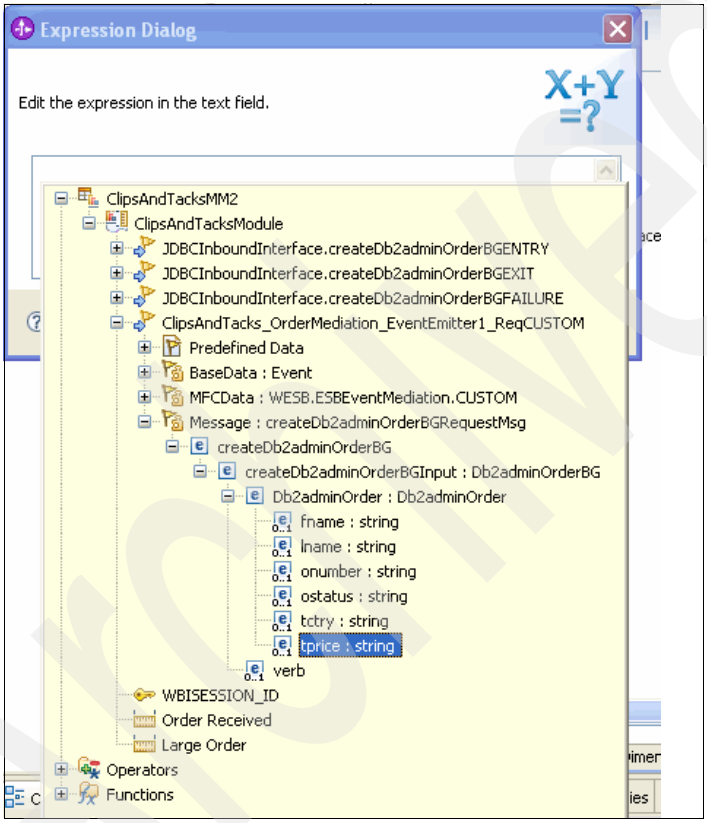


Figure 6-107 Expression Dialog

6. Save your work.

Adding the No. of Large Order KPI

The Large Order KPI measures the number of large orders that are received by ClipsAndTacks' and tracks the current status against a given target. To add the No. of Large Order KPI:

1. In the monitor model editor (Figure 6-108), click the **KPI Model** tab. Right-click **ClipsAndTacksKPIContext** and select **New** → **KPI**.

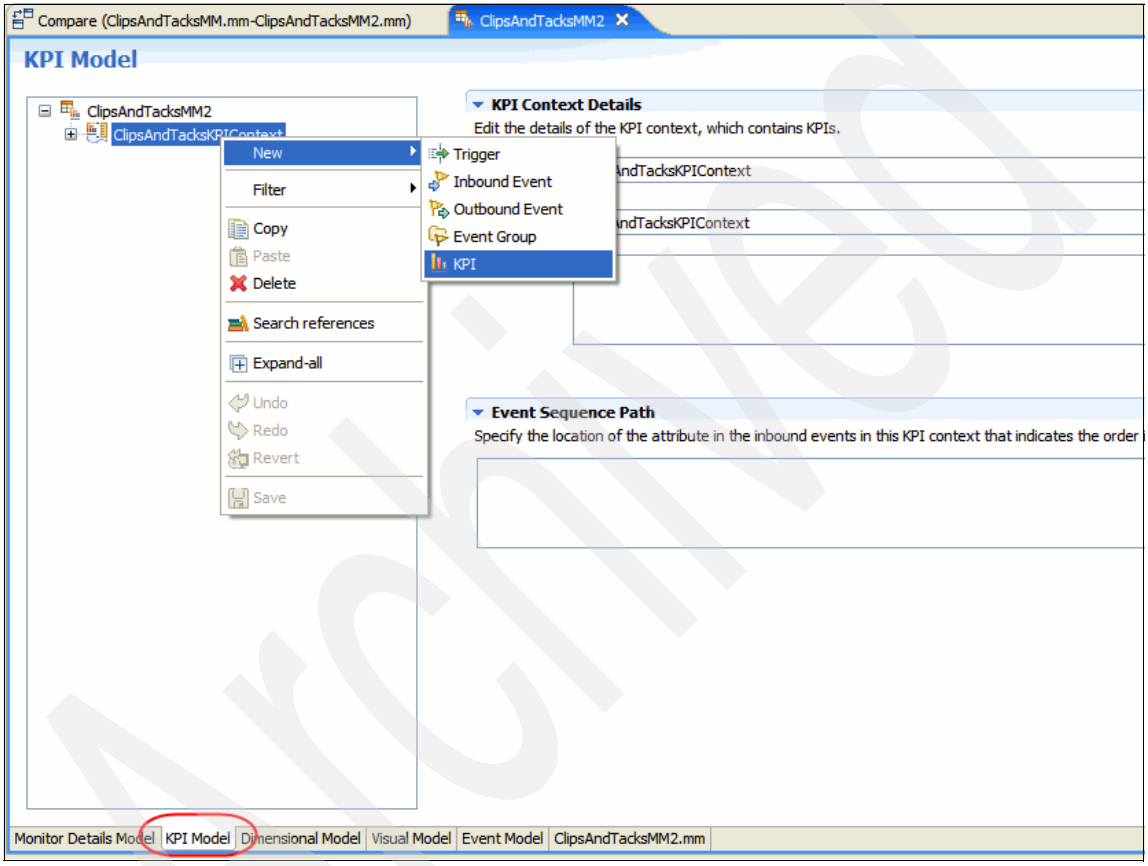


Figure 6-108 Creating a new KPI

2. For the name of the new KPI, type `No. of Large Orders Received`. Then click **OK**.

Archived

3. Under the KPI Target and Ranges section in the right panel (Figure 6-109), for Target, type 1. Click the **Add** button to add the Low, Satisfactory, and Excellent Ranges. Fill in the range values by using the values in Table 6-7.

Table 6-7 No. of Large Orders Received KPI target range

Range name	Start value	End value
Low	0	2
Satisfactory	2	4
Excellent	4	6

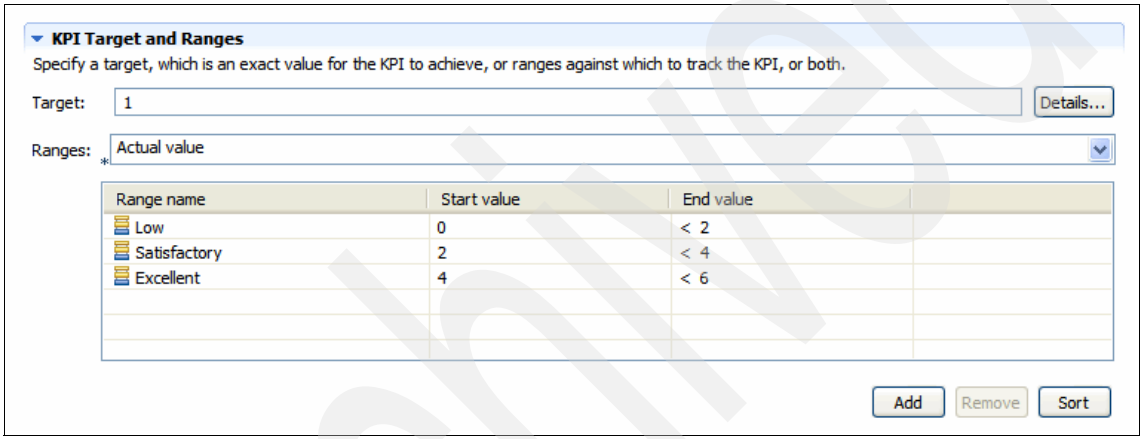


Figure 6-109 KPI Target and Ranges

4. Scroll down to the KPI Definition section (Figure 6-110), and select **Base this KPI on a metric and an aggregate function**.

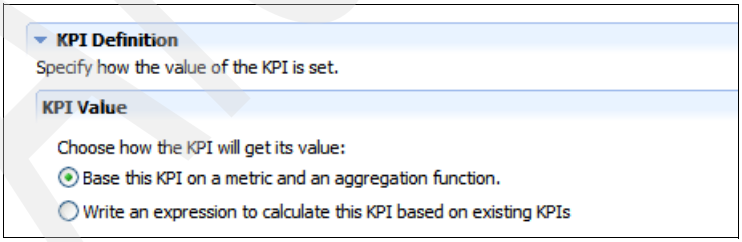


Figure 6-110 KPI Value

5. Under the KPI Details heading (Figure 6-111):
   a. For Monitoring Context, click **Browse**. Select **ClipsAndTacksModule**, and click **OK**.
   b. For Metric, click **Browse**. Select **Large Order** and click **OK**.
   c. For Aggregate Function, select **Count**.

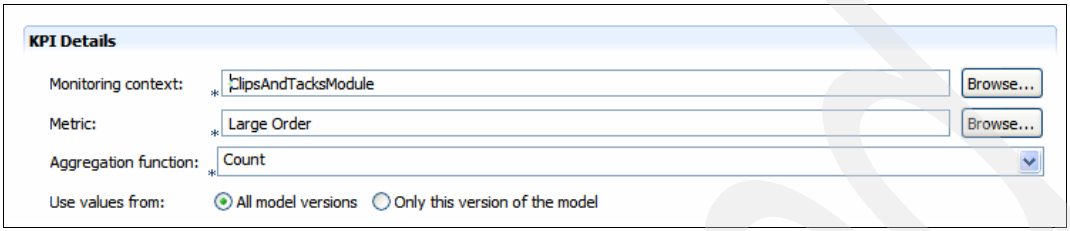


Figure 6-111 KPI Details

6. Save your work.

Generating the new Monitor J2EE project

To generate the new Monitor J2EE project:

1. In the Project Explorer view, right-click the edited monitor model, **ClipsAndTacksMM2.mm**, and select **Generate Monitor J2EE project**.
2. Accept the default for the target project names and click **Finish**.
3. After the project is generated, click **Finish** if prompted.
4. Click the **Server** tab.

Archived

5. Add the newly created Monitor project to the server.
   a. Right-click the server and select **Add and Remove Projects**.
   b. Add **ClipsAndTacksApp** and **ClipsAndTacksMM2Application** to the server. Click **Finish** (Figure 6-112 on page 339).

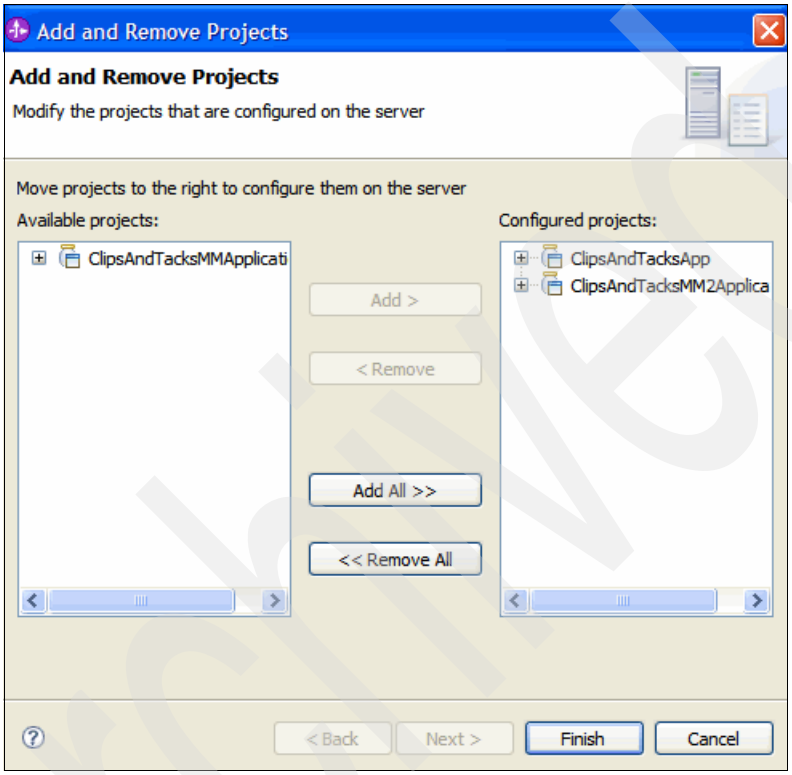


Figure 6-112 Adding the Monitor J2EE project to the server

Archive

Creating the dashboard

To create the dashboard:

1. Launch **WebSphere Business Monitor Dashboard**. The default user name and password are `admin`.
2. Click the **Dashboards** tab. In the OrderDashboard created earlier, click and drag the **KPI icon** to the dashboard to add a new KPI portlet (Figure 6-113).

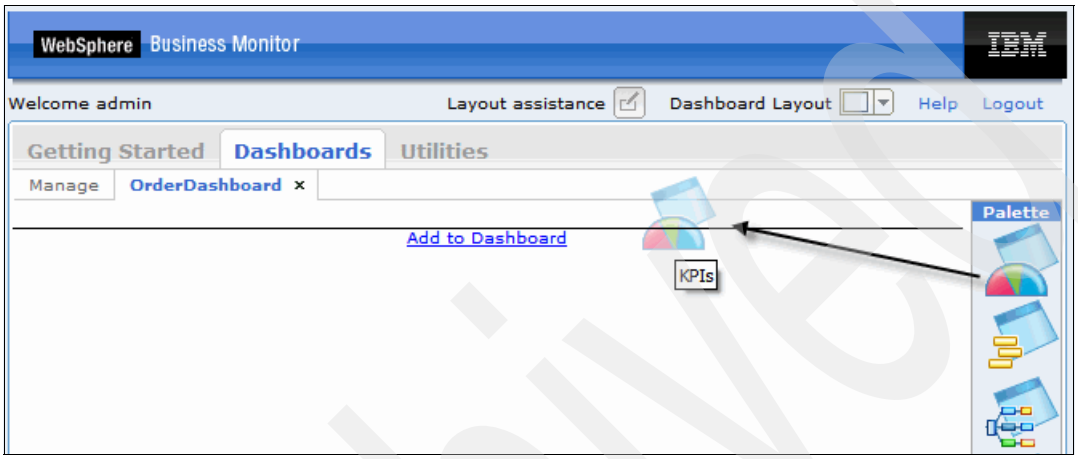


Figure 6-113 Adding the KPI portlet

3. Click **Personalize** and select the **No. of Large Orders Received** and **No. of Orders Received** KPIs (Figure 6-114). Click **OK**.

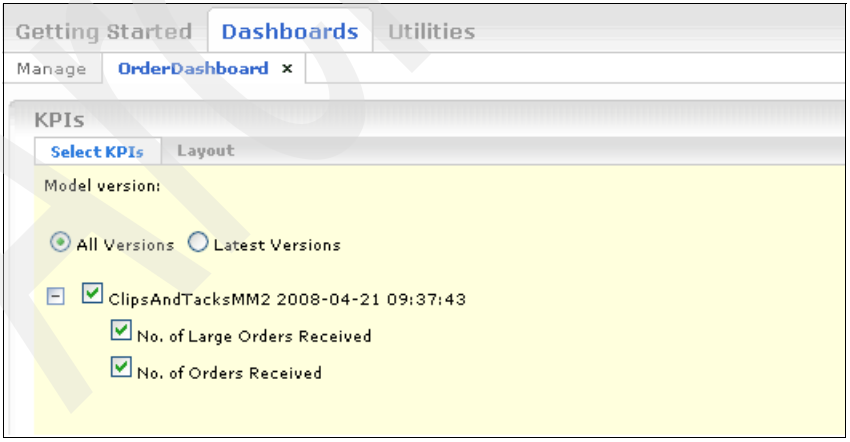


Figure 6-114 Adding orders received and large orders KPIs to the dashboard

Figure 6-115 shows the newly added KPI portlet.

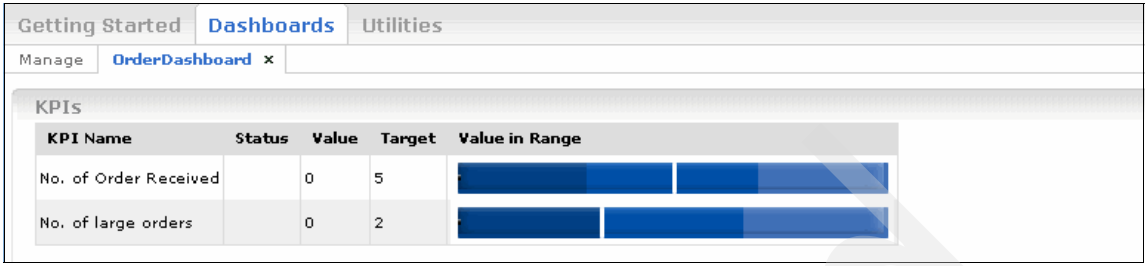


Figure 6-115 Newly added KPIs

6.5.3 Testing the Event Emitter primitive scenario

Ensure that the order records are deleted. In the DB2 Command Line Processor, run the **`LoadOrder.sql`** script (Example 6-3 on page 290). Figure 6-116 shows the KPI portlet after the dashboard updates if you completed the steps successfully.

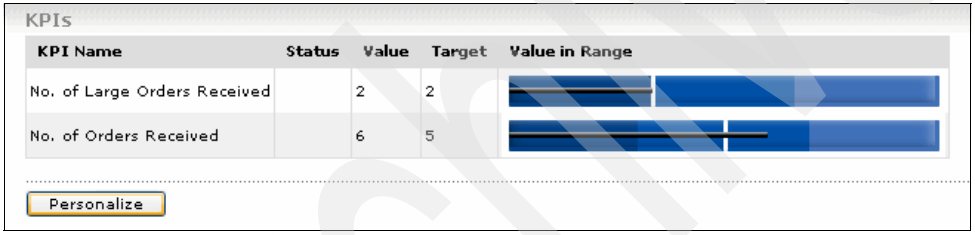


Figure 6-116 KPI result

Archive

6.6 Adding the Percentage of Large Orders KPI

In this section, we add an additional KPI, Percentage of Large Orders KPI, dynamically after the monitor model is deployed:

1. On the WebSphere Business Monitor dashboard, click the **Utilities** tab and and click the **KPI Manager** subtab (Figure 6-117). For Mode, select **ClipsAndTacksMM2** as the monitor model. Click the **Actions** button.

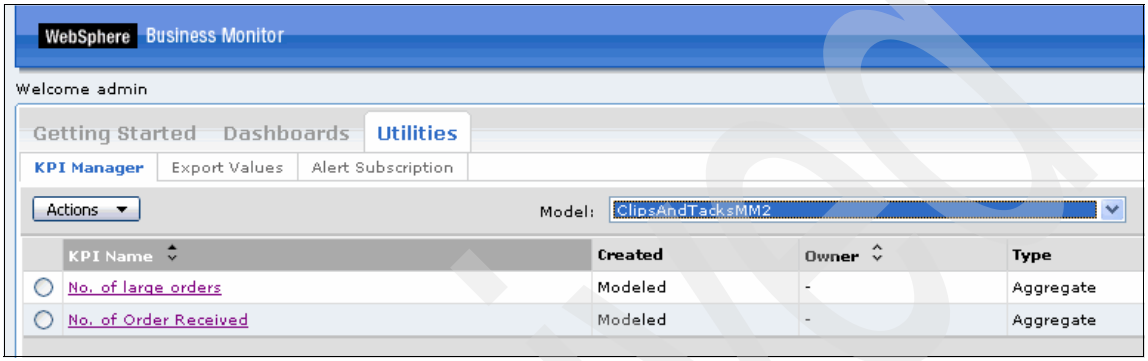


Figure 6-117 KPI Manager tab

2. Select **New Expression KPI**.
3. In the New Expression KPI Properties panel (Figure 6-118), click the **Name** tab. For KPI name, type `Percentage of large Orders`. Click **OK**.

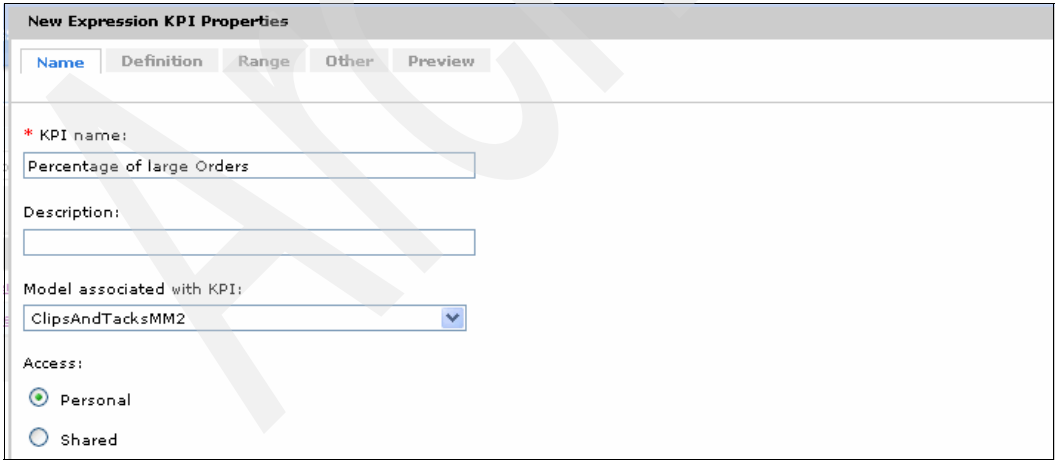


Figure 6-118 New Expression KPI Properties panel

4. Click the **Definition** tab (Figure 6-119). For Specify the expression that will define the KPI, type `(No._of_Large_Orders_Received div No._of_Orders_Received) * 100`. You can use the KPI and Operator drop-down lists to assist you. In the drop-down list, select the individual KPI and click **Insert**.

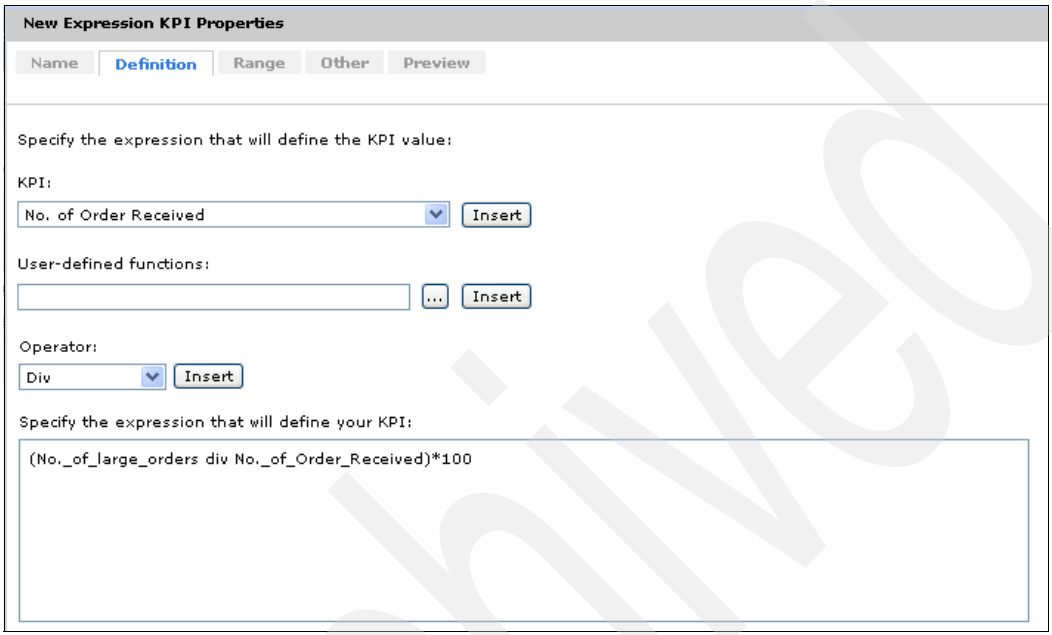


Figure 6-119 Expression KPI Definition

In the KPI Manager, the newly created Percentage of Large Orders KPI are listed.

5. Go back to the **Dashboard** tab. Select **OrderDashboard**. Select the newly created **Percentage of Large Orders** KPI and click **OK**.
6. Double-click the **Percentage of Large Orders** KPI.

Archived

7. In the KPI Properties panel (Figure 6-120), click the **Range** tab.

   a. Click **Add row**.
   b. For Target, type `20`.
   c. For Range Name, type `Low`.
   d. For the start value, type `0` and for the end value, type `15`.
   e. Click the color icon and select **Red**.
   f. Configure the KPIs by clicking the **Personalize** button.
   g. Do the same for the Satisfactory and Excellent range as shown in Table 6-8.

Table 6-8 Percentage of Large Orders Received KPI target range

Range name	Start value	End value	Color
Low	0	15	Red
Satisfactory	15	75	Orange
Excellent	75	100	Green

You can also add icons for the KPI by clicking the space under the Icon column.

   h. Click **OK**.

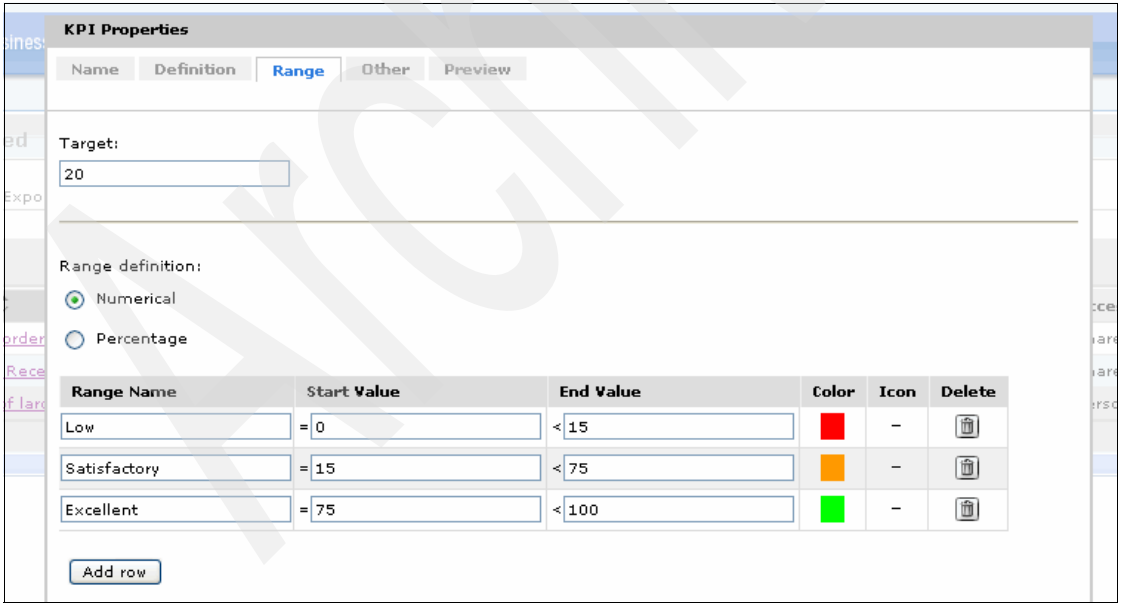


Figure 6-120 KPI target and ranges

8. Return to the **Dashboard** tab. Click **Personalize** on the KPI portlet created previously.
9. Select the newly created KPI as shown in Figure 6-121 and click **OK**.

Figure 6-121 Adding the newly created KPI

Figure 6-122 shows the completed KPI portlet, with the newly added Percentage of Large Orders KPI.

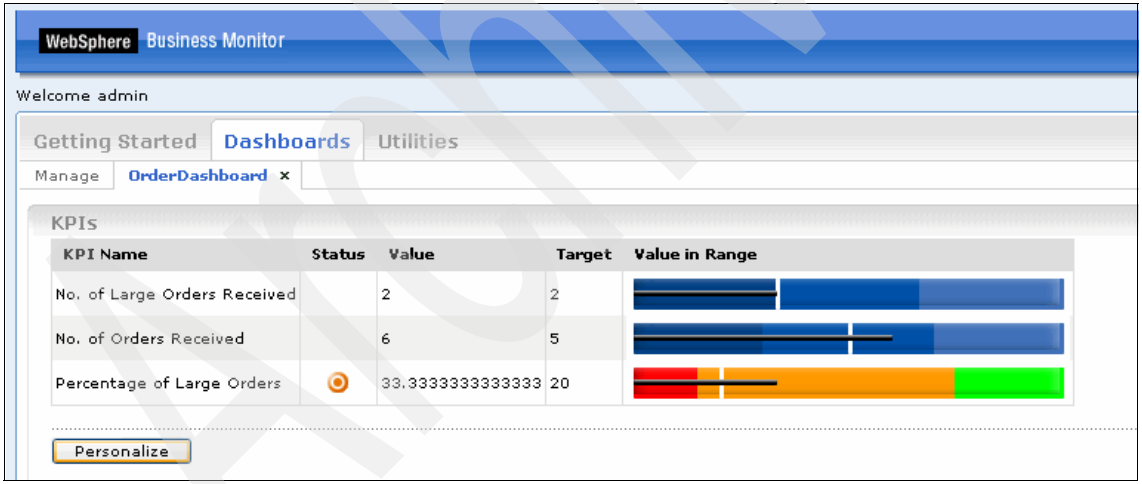


Figure 6-122 Completed KPI portlet

Archived

6.6.1 Cleaning up the dashboard

After you add the Percentage of Large Orders KPI, remove the KPIs from the dashboard:

1. Click the menu icon in the top right corner of the KPI portlet and select **Delete**.
2. Remove all projects from the server. In the Server view, right-click the **WebSphere Business Monitor Server v6.1 on WebSphere ESB** server and select **Add and Remove Projects**.
3. In the Add and Remove Projects window, click **Remove All** to remove all projects and click **Finish**.
4. Stop the server.
5. Connect to the ORDER database again and delete the records from the tables (Example 6-4 on page 291).

6.6.2 Troubleshooting: NameNotFoundException error messages

If you encounter the NameNotFoundException error message (Example 6-5), perform the steps that follow to resolve it:

Example 6-5 NameNotFoundException

```
0000003b EventBusSende E com.ibm.events.emitter.impl.EventBusSender
initialize CEIEM0020E The emitter failed to initialize because the JNDI
lookup on the event bus home name failed.
JNDI name: ejb/com/ibm/events/bus/EventBus
Context: WBMonSrv_esb_Cell/nodes/WBMonSrv_esb_Node/servers/server1
Exception (if any): javax.naming.NameNotFoundException
```

1. In the Servers view, right-click **WebSphere Business Monitor Server v6.1 on WebSphere ESB** and select **Add and Remove Projects**.
2. In the Add and Remove Projects window, click **Remove All** and click **Finish**.
3. Click **OK** when prompted.
4. Right-click **WebSphere Business Monitor Server v6.1 on WebSphere ESB** and select **Restart** → **Start**.
5. From the menu bar, select **Project** → **Clean**.
6. Select **Clean All Projects** and click **OK**.

Archived

7. Go back to the Servers view, right-click **WebSphere Business Monitor Server v6.1 on WebSphere ESB** and select **Add and Remove Projects**.
8. In the Add and Remove Projects window, click **Add All** and click **Finish**.
9. Click **OK** when prompted.

6.7 Design considerations

In this section, we discuss the following design considerations:

- When to monitor events from the event sources directly
- When to use built-in service component events and Event Emitter primitive
- Where to use the Event Emitter primitive in the mediation flow

6.7.1 When to monitor events from the event sources directly

In situations where an event source is not managed by a process engine, such as WebSphere Process Server, event sources can be monitored directly. This can occur by using a combination of adapters and ESB technologies, as described in this chapter, or through custom event emitters, as described in Chapter 9, “Custom event emitter” on page 599. However in situations where the event source is managed by a process engine, the choice might not be as straightforward. The alternative approach in such situations is to monitor the events that are emitted from the process engines as the flows are executed. These events can be emitted when there is a task in the process flows that updates the EIS.

One of the design principles you can adopt is to monitor events from the location that provides the most complete source of information that will give you an accurate view of the business measures that you want to capture.

In the fictitious scenario described here, we chose to monitor events from the event source (Order Database) directly. If we were only interested in monitoring the updates to the database from WebSphere Process Server and WebSphere MQ Workflow, we could have chosen to monitor the events generated from the process engines instead.

However, in our scenario, besides WebSphere Process Server and WebSphere MQ Workflow, other applications have access to the Order Database. Hence, we have taken the approach of monitoring the database directly.

6.7.2 When to use built-in service component events and Event Emitter primitive

In 6.4, “Scenario 1: Monitoring the built-in service component events” on page 258, we discuss the use of SCA event natures, which are ENTRY, EXIT, and FAILURE, and component specific event natures. In 6.5, “Scenario 2: Monitoring mediation flow custom events” on page 311, we present the example where the Event Emitter primitive is used. However, we have not addressed the frequently asked question: When do I use the Event Emitter primitive or the built-in service component events?

The Event Emitter primitive offers the ability to emit events based on the content of the data as they are processed in the mediation flow. For example, by using the Event Emitter primitives, you can filter the events from the event sources and only emit events that are of significance to the business measures that you are trying to capture.

However, if no requirements exist to monitor arbitrary events during the execution of the mediation flow, monitoring the event natures of the service components may be sufficient. Conversely, if we are only interested in monitoring events that occur during the execution of a mediation flow, service component monitoring is not required.

6.7.3 Where to use the Event Emitter primitive in the mediation flow

The Event Emitter primitive provides a way to generate significant business events from within a mediation flow. Think of an Event Emitter primitive as a notification mechanism that is used to indicate an unusual event, such as a significant failure within a flow or an unusual path run in the flow. Avoid placing an Event Emitter primitive in the normal path of a flow because this can affect performance by causing a large number of events to be generated.

In addition, consider the data that you choose to store in the event. The data has an effect on the system performance as well. Store only data that is relevant for your context.

Examples where an Event Emitter primitive may not be appropriate

The example in Figure 6-123 on page 349 shows the Event Emitter placed before the Message Filter. This means that events are emitted every time the flow is executed. This placement has an unnecessary performance impact on the system if we are only interested in monitoring orders that large.

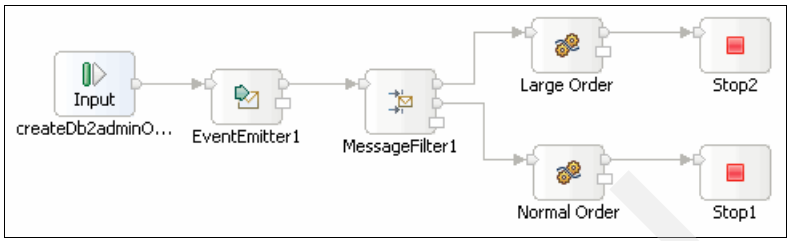


Figure 6-123 Placing an Event Emitter in the main path

In the ClipsAndTacks Office Supplies Pte. Ltd. scenario, most of the orders are classified as normal and take the default branch. Assuming that the event emitters were placed as shown in Figure 6-124, events are emitted most of the time. This design is not desirable because it impacts the performance of the system.

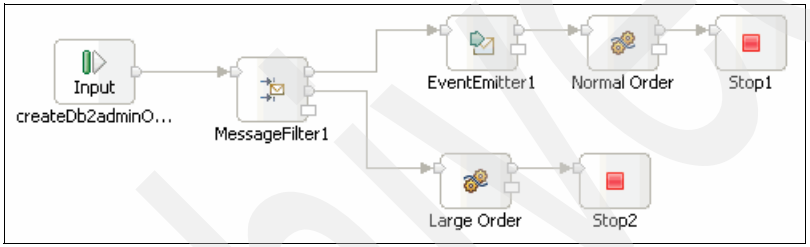


Figure 6-124 Placing an Event Emitter in the normal path

In most cases, for BAM, you are interested in monitoring the occurrence of specific events. Therefore, capturing the occurrence of normal events may not be necessary. It may be meaningful to review your design to avoid the emission of events in the normal path to avoid this where possible.

Example where an Event Emitter primitive can be used

The flow in Figure 6-125 shows an Event Emitter primitive used when there is an exceptional or unusual event that occurs in the flow.

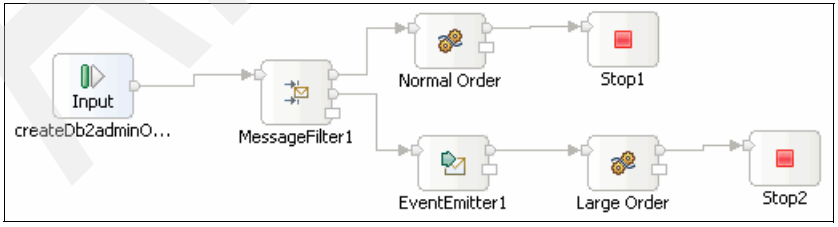


Figure 6-125 Event Emitter primitive used where an unusual event occurs

An event emitter can also be used when a failure occurs in the message flow.

6.8 Summary

In this chapter, we have seen how WebSphere Adapters and WebSphere Enterprise Service Bus can be employed together to retrieve business events from EISs and emit events in the Common Base Event format to WebSphere Business Monitor. We have also explored the concept of service component monitoring and the generation of custom events through the use of a fictitious business scenario.

Archived

7

FileNet P8 BPM

In this chapter, we explain how to integrate FileNet P8 BPM with WebSphere Business Monitor. We define all the relevant information about FileNet P8 BPM Version 4.0.2 process elements that are monitored by WebSphere Business Monitor Version 6.1.1 to verify the business goals.

Archived

© Copyright IBM Corp. 2008. All rights reserved.

7.1 Introduction to FileNet P8 BPM

The IBM FileNet P8 BPM Suite provides a comprehensive platform for automating, integrating and optimizing an organization's business processes. It is an essential part of IBM FileNet P8 Enterprise Content Management (ECM) platform, which provides extensive content management and compliance solutions. Figure 7-1 shows the logical architecture of IBM FileNet P8 and where IBM FileNet P8 BPM fits within it.

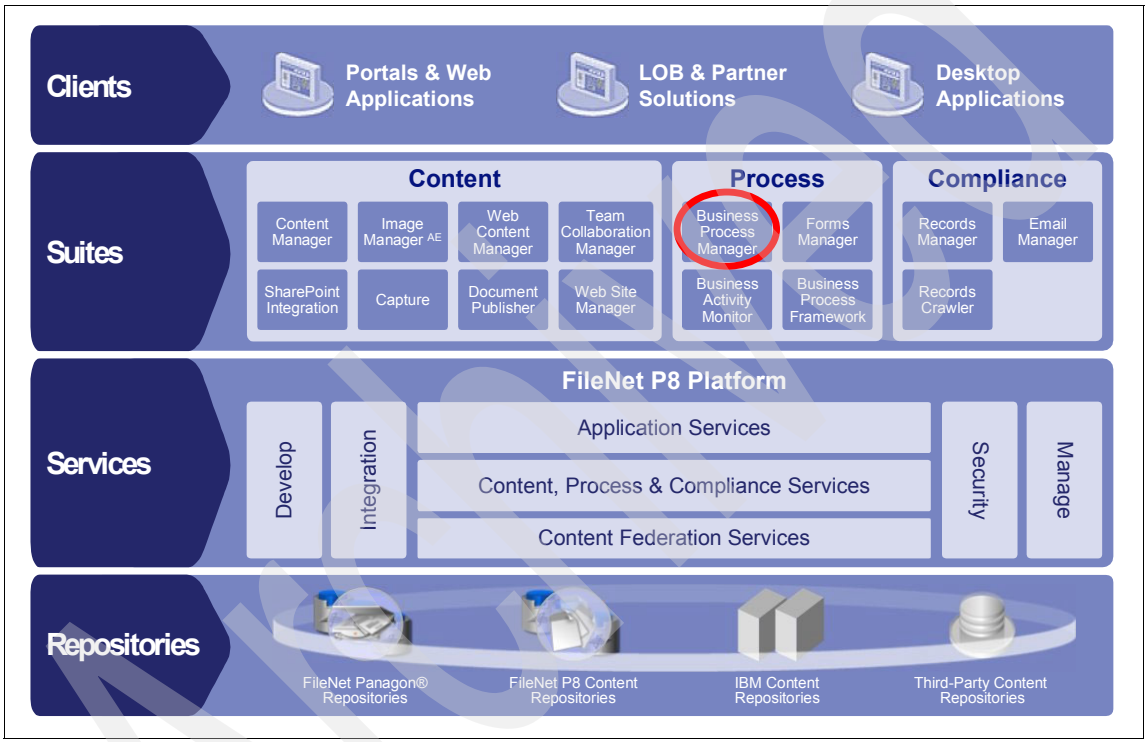


Figure 7-1 Logical diagram of the IBM FileNet P8 ECM platform and IBM FileNet BPM

FileNet P8 BPM manages workflow among people and systems for content and human-centric processes. It provides the following functions:

- Comprehensive process management including process modeling, advanced analytic and simulation and business activity monitoring
- Automatic management of process exceptions to allow agile, immediate responses to internal events or customer demands
- FileNet P8 integration with other IBM ECM solutions

- Process standards, such as process modeling (BPMN), and definition and execution (XML Process Definition Language (XPDL))
- Automation through full process modeling and design so that repetitive tasks are automated to boost productivity and reduce cost
- Optimization by analysis and simulation

FileNet P8 BPM is fully integrated with the FileNet P8 platform. By using the FileNet concept of active content, business processes can be executed based on changes in the metadata of documents, changes in the state of a document's life cycle, or whenever a document is added to the IBM FileNet P8 platform.

FileNet P8 BPM combines the concepts of *process* with *active content management*. This combination facilitates a variety of process management scenarios. Active content can represent literally anything that is of interest to the business, from unstructured documents to large object (LOB) applications and customer cases. Each piece of content has metadata that can include information about its creation date, owner, keywords, and so on. Any change in that content or its metadata is an event that can be captured and acted upon.

For example, such a change can cause a process to launch. The process in its execution may, in turn, cause other changes to the content.

Content provides a context for a process in which better decisions can be made. Furthermore, metadata, such as process, rule, and forms definitions (in XML or other formats), are themselves content that can and should be managed as such.

FileNet P8 uses a publish-and-subscribe event-based model. As business objects in the repository change state (created, modified, or deleted), the system can automatically launch the appropriate process or make a behavioral change in an in-flight process.

FileNet P8 BPM also provides various Web services capabilities, such as the ability to consume Web services in processes and publish processes as Web services, as well as Web services (Web Services Description Language (WSDL) based) APIs. These enable IBM FileNet P8 BPM to support service-oriented architecture (SOA) initiatives and interoperate in an SOA environment.

Archive

7.2 Overview of integrating FileNet P8 BPM with WebSphere Business Monitor

You can use IBM WebSphere Business Monitor to monitor active FileNet P8 BPM processes. FileNet P8 BPM software includes a Common Base Event Adapter, a basic business process management (BPM) monitor model, and basic dashboard views that you configure and deploy to the WebSphere Application Server and WebSphere Business Monitor server. Figure 7-2 shows the main components of the integration.

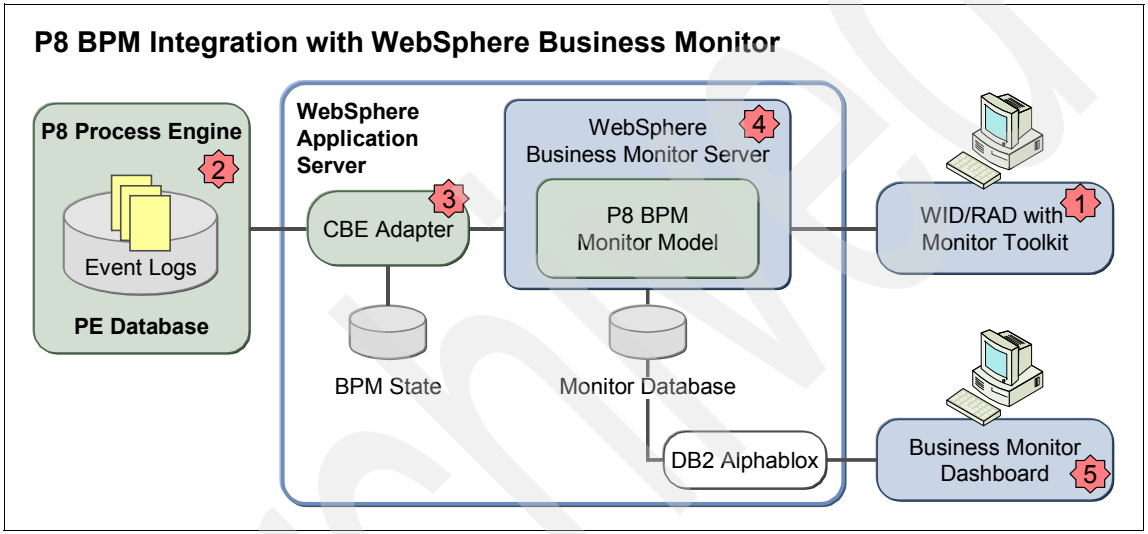


Figure 7-2 FileNet P8 BPM Integration with WebSphere Business Monitor

The numbers in the following sequence correspond to those in Figure 7-2:

1. A Monitor Model developer uses WebSphere Integration Developer or Rational Application Developer with Monitor Toolkit to configure the BPM Monitor Model and add user-defined fields and key performance indicators (KPIs) that will be monitored.
2. At run time, the process engine records events that are associated with running processes, such as the start and end of each process and activity, in event logs.
3. The Common Base Event Adapter reads events from the process engine event logs, transforms the events to the Common Base Event format, and sends the Common Base Events to the WebSphere Business Monitor server.
4. The Business Monitor server processes and analyzes events and generates statistics.

5. The Business Monitor Dashboard provides customizable graphical views of data that is produced by running processes. It includes tools to analyze trends and generate alerts based on KPIs.

7.3 Runtime topology for the sample

In this sample, we design two machine topologies for the runtime environment as shown in Figure 7-3:

- The software installed on Machine 1:
  - FileNet P8 Process Engine Version V4.0.2
  - Microsoft SQL Enterprise 2000 SP4 (for the process engine database)
  - Microsoft Active Directory® Server (for user LDAP)
- The software installed on Machine 2:
  - WebSphere Business Monitor V6.1.1
  - DB2 Enterprise Edition V9.1
  - FileNet P8 BPM Common Base Event Adapter

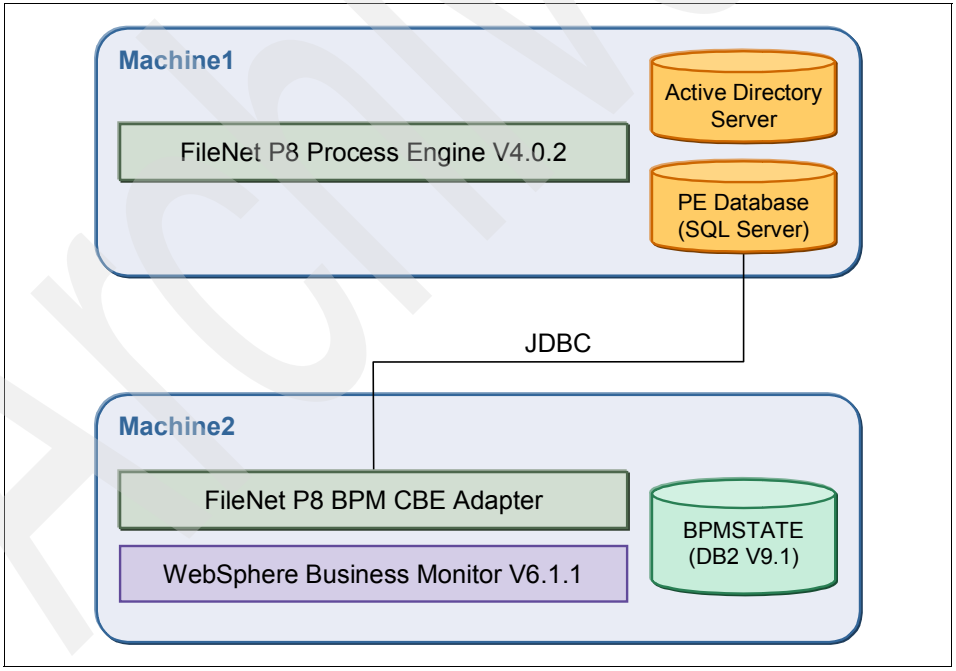


Figure 7-3 Runtime topology of the sample

Archived

Figure 7-4 illustrates the Monitor Model development topology. We use WebSphere Integration Developer V6.1 as the development platform.

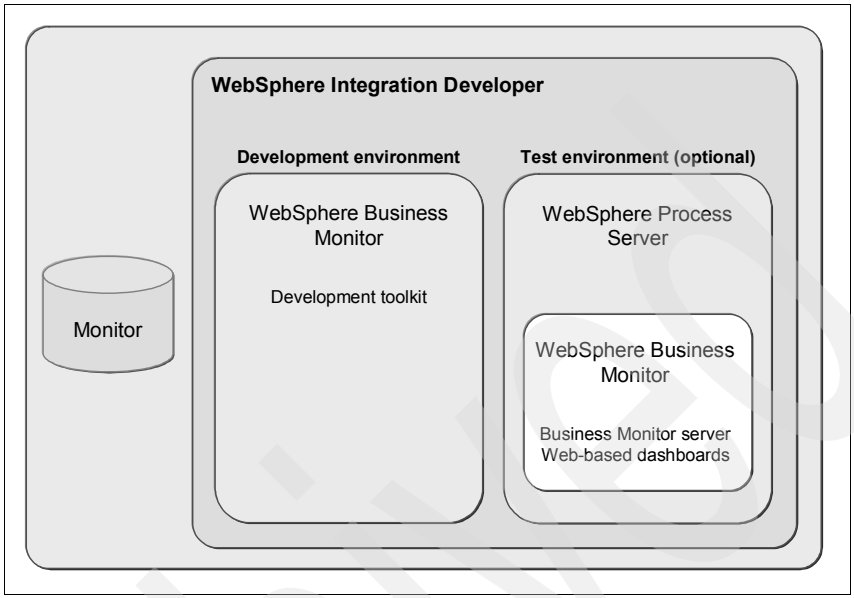


Figure 7-4 Development topology of the sample

Note: In this case, we assume that WebSphere Business Monitor V6.1.1 and the toolkit are installed and configured as explained in Chapter 3, "Using WebSphere Business Monitor" on page 55. We also assume that FileNet P8 and Microsoft Active Directory Server are installed. In addition, we assume that DB2 Enterprise Server Edition V9.1 is installed on the WebSphere Business Monitor machine.

The FileNet P8 BPM Common Base Event Adapter is supplied in the FileNet P8 Fix Pack 4.0.2 that is available for download from the following Web address:

http://www-1.ibm.com/support/docview.wss?rs=3247&context=SSTHRT&context=SSS236&context=SSNW2F&context=SS5UVB&context=SS5UUH&dc=D400&uid=swg24016610&loc=en_US&cs=UTF-8&lang=en

7.4 Deploying the FileNet P8 BPM Common Base Event Adapter to the WebSphere Business Monitor server

The Common Based Event Adapter component links together the process engine and the WebSphere Business Monitor server. Figure 7-5 illustrates how the FileNet P8 BPM Adapter works. When processes are running, the process engine records each event to the event logs in the process engine database. The FileNet P8 BPM Common Base Event Adapter for WebSphere reads this data, converts the process engine events to Common Base Event events, and sends the converted events to the WebSphere Business Monitor server through the Common Event Infrastructure (CEI) layer. Then the state is recorded to the Common Base Event Adapter Database.

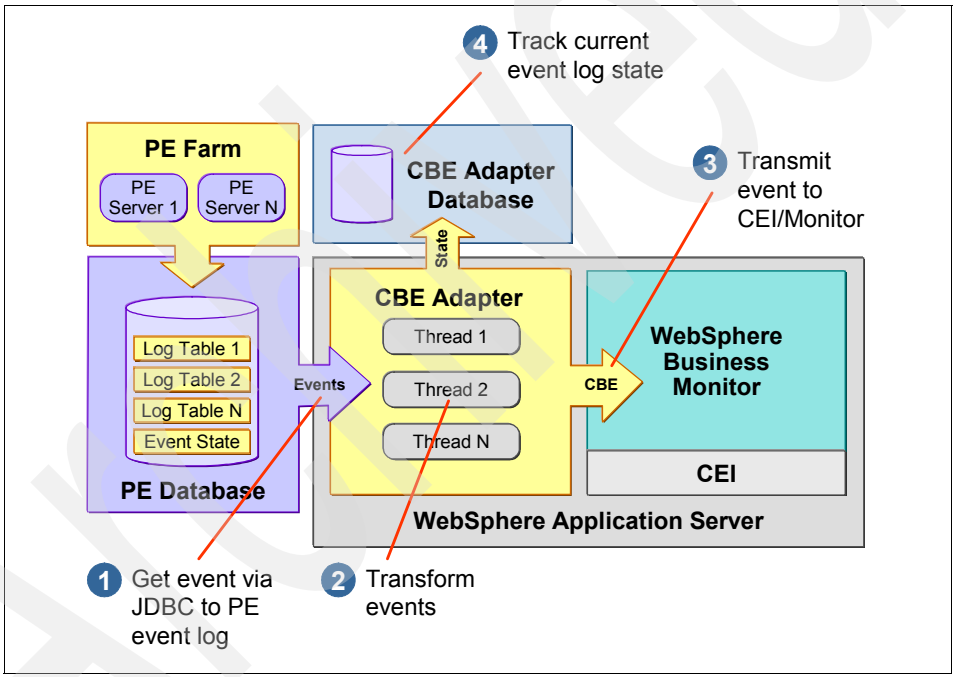


Figure 7-5 FileNet P8 BPM Common Base Event Adapter

In the sections that follow, we explain how to deploy and configure the FileNet P8 BPM Common Base Event Adapter for WebSphere before you run the WebSphere Business Monitor. Table 7-1 on page 358 outlines the steps.

Table 7-1 Steps for deploying the FileNet P8 BPM Adapter

Step	Description
Step 1: Create the BPMSTATE database.	Execute the P8BPMCBEAdaptor.ddl script, which creates the BPMSTATE database and tables used by the BPM Common Base Event Adapter for WebSphere.
Step 2: Create a work manager.	Create a work manager, which is used to query events from multiple process engine event logs at the same time.
Step 3: Create the database authentication aliases.	Create authentication aliases to connect to the process engine and Common Base Event Adapter databases.
Step 4: Create a JDBC data source.	Create a JDBC data source to connect to the process engine database and Common Base Event Adapter databases.
Step 5: Install the Common Base Event Adapter Application.	Browse to the P8BPMCBEAdaptor.ear file and install the Common Base Event Adapter.
Step 6: Set the optional configuration properties.	Set the optimal configuration properties to fine-tune the behavior of the Common Base Event Adapter.
Step 7: Start the BPM Common Base Event Adapter for WebSphere.	Start the FileNet P8 BPM Common Base Event Adapter for WebSphere.

7.4.1 Creating the BPMSTATE database

To execute the P8BPMCBEAdaptor.ddl script to create the BPMSTATE database and tables used by the FileNet P8 BPM Common Base Event Adapter for WebSphere:

1. Log in to the FileNet P8 BPM Common Base Event Adapter server as a user with the necessary DB2 privileges to create the new BPMSTATE database.
2. Open a DB2 command window.
3. Change to the directory that contains the P8BPMCBEAdaptor.ddl script.
4. In the DB2 command window, type the following command:

```
db2 +c -stvf P8BPMCBEAdaptor.ddl
```

7.4.2 Creating the work manager

A *work manager* is a thread pool used by a WebSphere application to perform tasks asynchronously. The Common Base Event Adapter uses a work manager to query events from multiple process engine event logs at the same time. Create a work manager for the Common Base Event Adapter to use:

1. To launch the WebSphere Administrative Console of the WebSphere Business Monitor machine, open the browser and type the following URL:

   `http://[WebSphere Business Monitor host]:9060/admin`

2. Log in as `admin`.

3. In WebSphere Administrative Console (Figure 7-6), expand **Resources** → **Asynchronous Beans** and click **Work managers**.

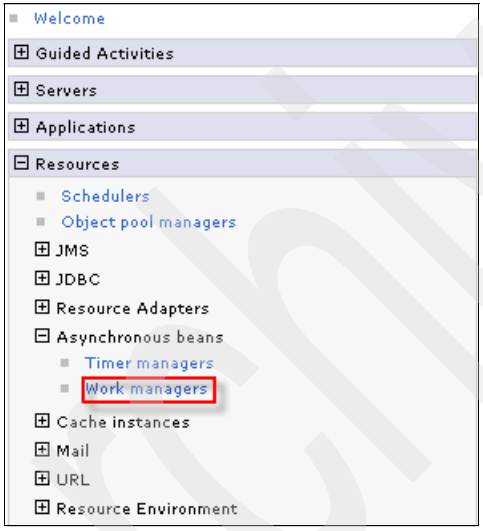


Figure 7-6 Navigating to Work managers

Archived

4. In the Work managers panel (Figure 7-7):
   a. For Scope, select **Node=<host name>Node01**. In our example, we use **Node=monitor02Node01**.
   b. Click **New** to create a new work manager.

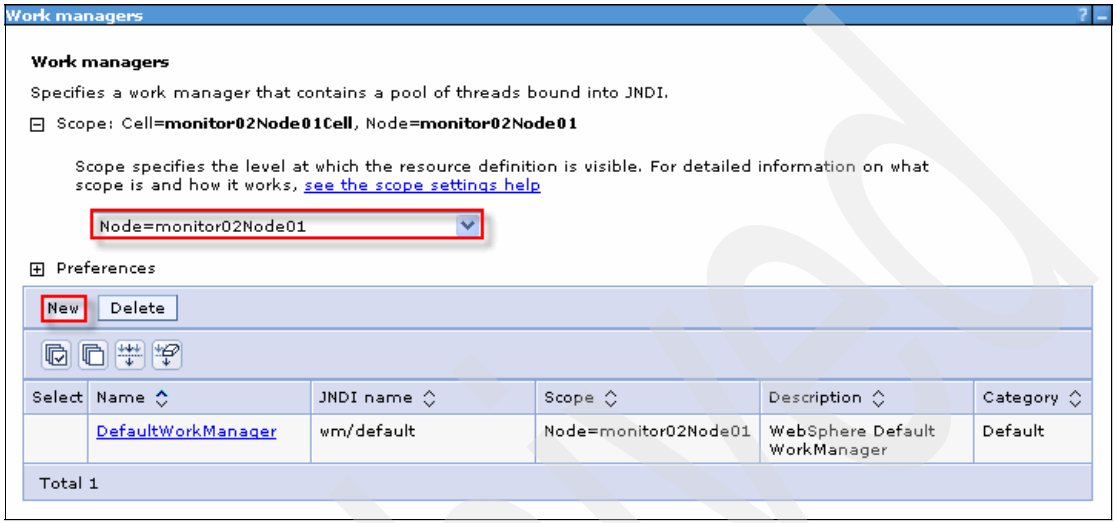


Figure 7-7 Work managers panel

5. In the General Properties section (Figure 7-8 on page 361):
   a. For Name, type `P8 BPM CBE Adaptor Work Manager`.
   b. For JNDI name, type `wm/com/ibm/P8/BPM/CBEAdaptor`.

> **Syntax:** Ensure that you type `CBEAdaptor`, not CBEAdapter.

   c. Clear the **Growable** check box because in this case we do not need the maximum number of threads that can be exceeded temporarily.
   d. Click **OK**.

Archived

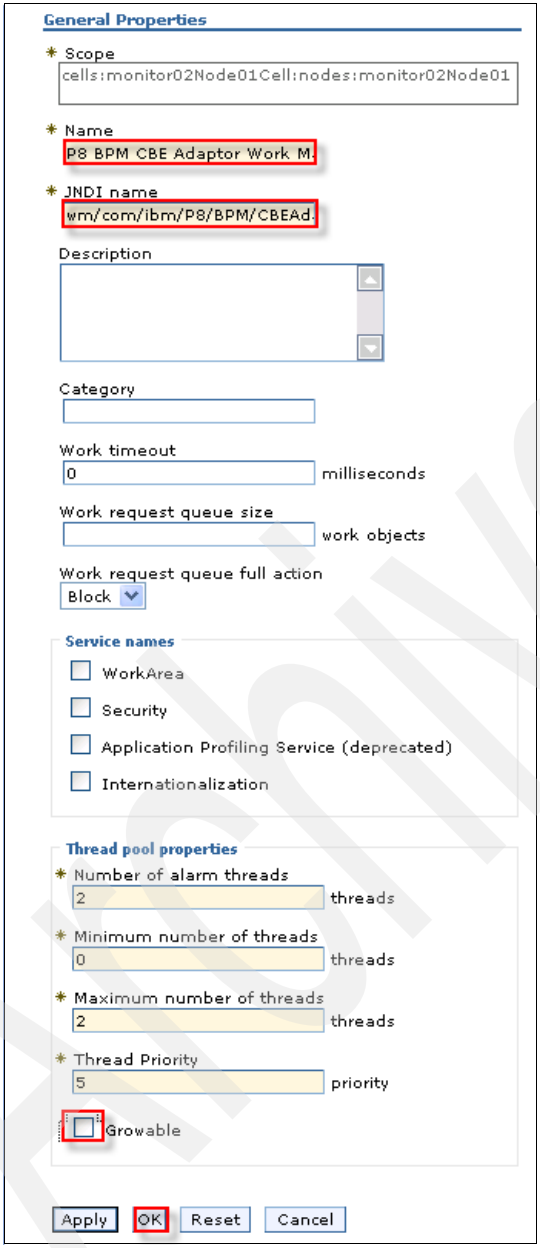


Figure 7-8 Work managers configuration

Archived

6. In the Messages box (Figure 7-9), click the **Save** directly to the master configuration link.

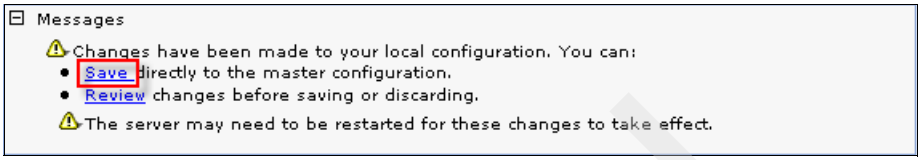


Figure 7-9 Saving directly to the master configuration

7.4.3 Creating the database authentication aliases

To create authentication aliases to connect to the process engine and FileNet P8 BPM Common Base Event Adapter databases:

1. In the WebSphere Administrative Console menu (Figure 7-10), expand **Security** and click **Secure administration, applications, and infrastructure**.

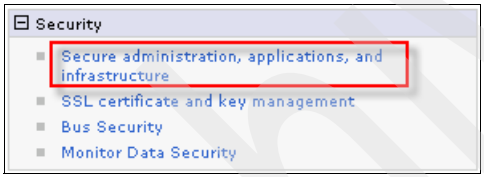


Figure 7-10 Security

Archived

2. In the Secure administration, applications, and infrastructure panel (Figure 7-11), under Authentication, expand **Java Authentication and Authorization Service** and select **J2C authentication data**.

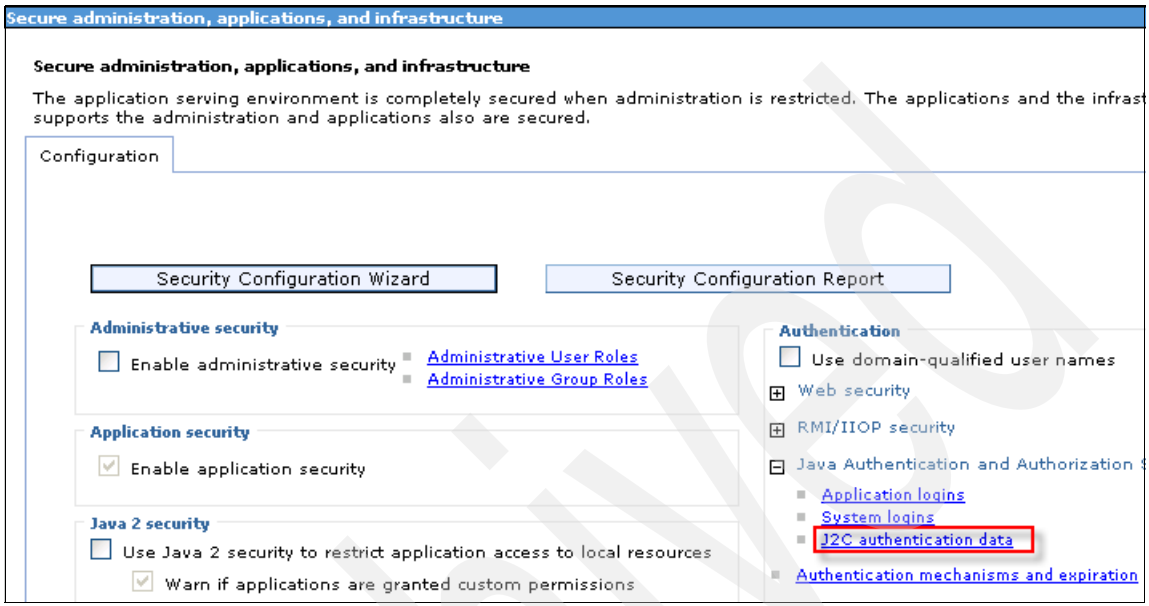


Figure 7-11 Secure administration, applications, and infrastructure

3. Create an authentication alias for use by the FileNet P8 BPM Common Base Event Adapter to connect to the process engine database. The user referenced by this alias must have privileges that allow the Common Base Event Adapter to query the process engine event log views and to update the VWPAEventState table. In our example, we use user `sa` and password `filenet`.

Archive

a. In the Preferences panel (Figure 7-12), click **New** to create an alias.

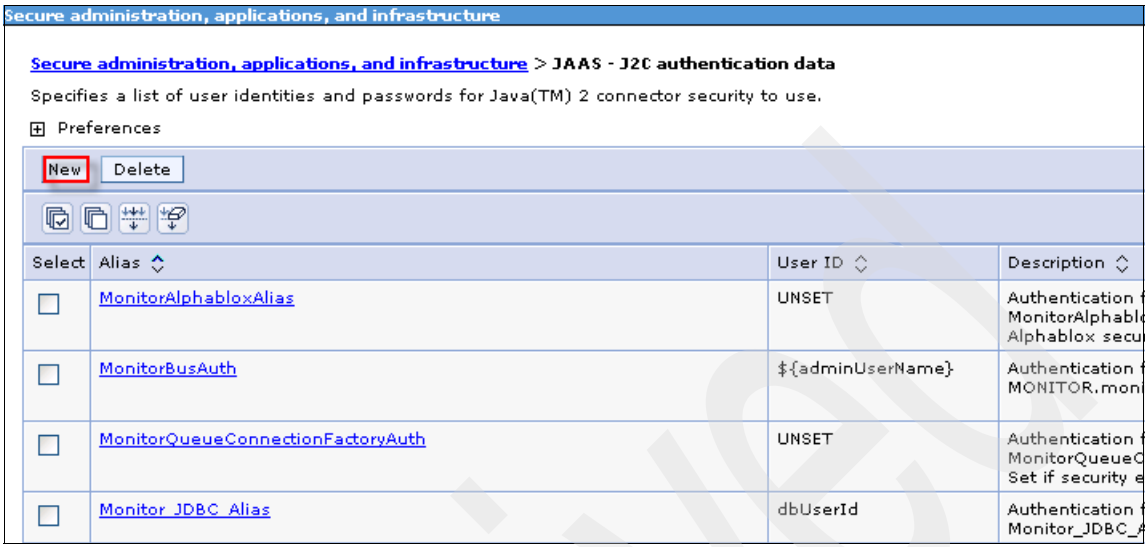


Figure 7-12 Secure administration, applications, and infrastructure panel

Archive

b. Under General Properties of the Configuration page (Figure 7-13):

i. For Alias, type `PEDBUser`.
ii. For User ID (process engine database user), type `sa`.
iii. For Password (password of process engine database user), type `filenet`.
iv. For Description, type `J2C Authentication data for VWdb database`.
v. Click **OK**.

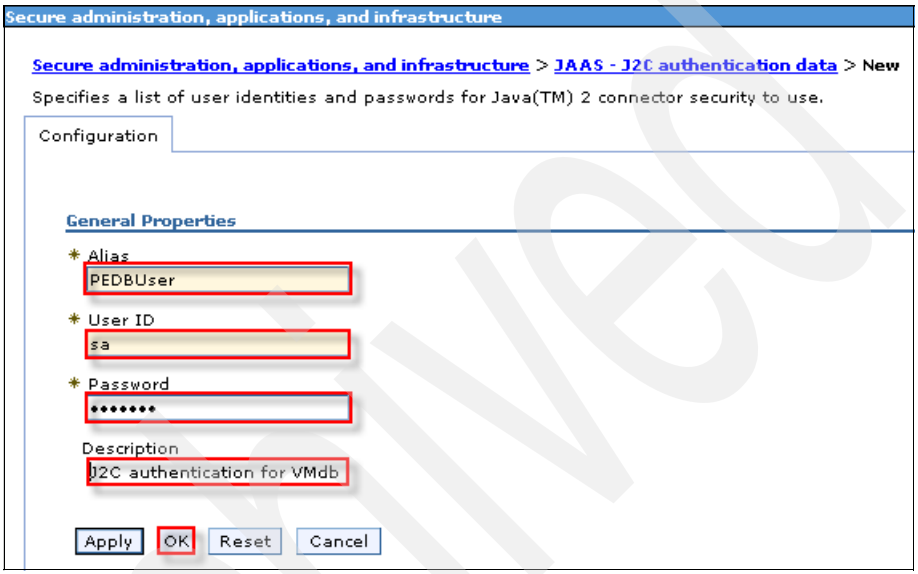


Figure 7-13 JAAS-J2C authentication data

c. Click **Save directly to the monitor configuration**.

4. Create an authentication alias for use by the FileNet P8 BPM Common Base Event Adapter for WebSphere to connect to the BPMSTATE database. The user referenced by this alias must have privileges allowing the Common Base Event Adapter to update the tables within the BPMSTATE database.

a. In the Preferences panel, click **New** to create an alias.

b. Under General Properties of the Configuration page:

i. For Alias, type `BPMSTATE_User`.
ii. For User ID (BPMSTATE database user), type `db2admin`.
iii. For Password (password of BPMSTATE database user), type `db2admin`.
iv. For Description, type `J2C Authentication data for BPMSTATE database`.
v. Click **OK**.

c. Click **Save** directly to the monitor configuration.

Archived

7.4.4 Creating the JDBC data sources

To create the JDBC data sources:

1. Create a JDBC Provider for connecting to the process engine database:
   a. In the WebSphere Administrative Console, expand **Resources** → **JDBC** and click **JDBC Providers**.
   b. In the Create a new JDBC Provider panel, for Scope, select **Node=<host name>Node01**. In our example, we select **Node=monitor02Node01**. Click **New**.
   c. In the Create new JDBC provider panel (Figure 7-14 on page 367):
      i. For Database type, select one of the following types. In this example, we select **SQL Server**:
         - DB2
         - Derby
         - Informix®
         - Oracle
         - Sybase
         - SQL Server
         - User-defined
      ii. For Provider type, select one of the following types. In this example, we select **SQL Server** for WebSphere embedded ConnectionJDBC driver:
         - **SQL Server**: WebSphere embedded ConnectionJDBC driver for MS SQL Server
         - **DB2**: DB2 Universal JDBC Driver Provider
         - **Oracle**: Oracle JDBC Driver

           Obtain the Oracle JDBC driver.JAR file. ojdbc14.jar. After you have this file, update the following WebSphere Class path environment variable to point to the.JAR file:

           ```
           ${ORACLE_JDBC_DRIVER_PATH}/ojdbc14.jar
           ```
      iii. For Implementation type, select **Connection pool data source**.
      iv. Click **Next**.

Archived

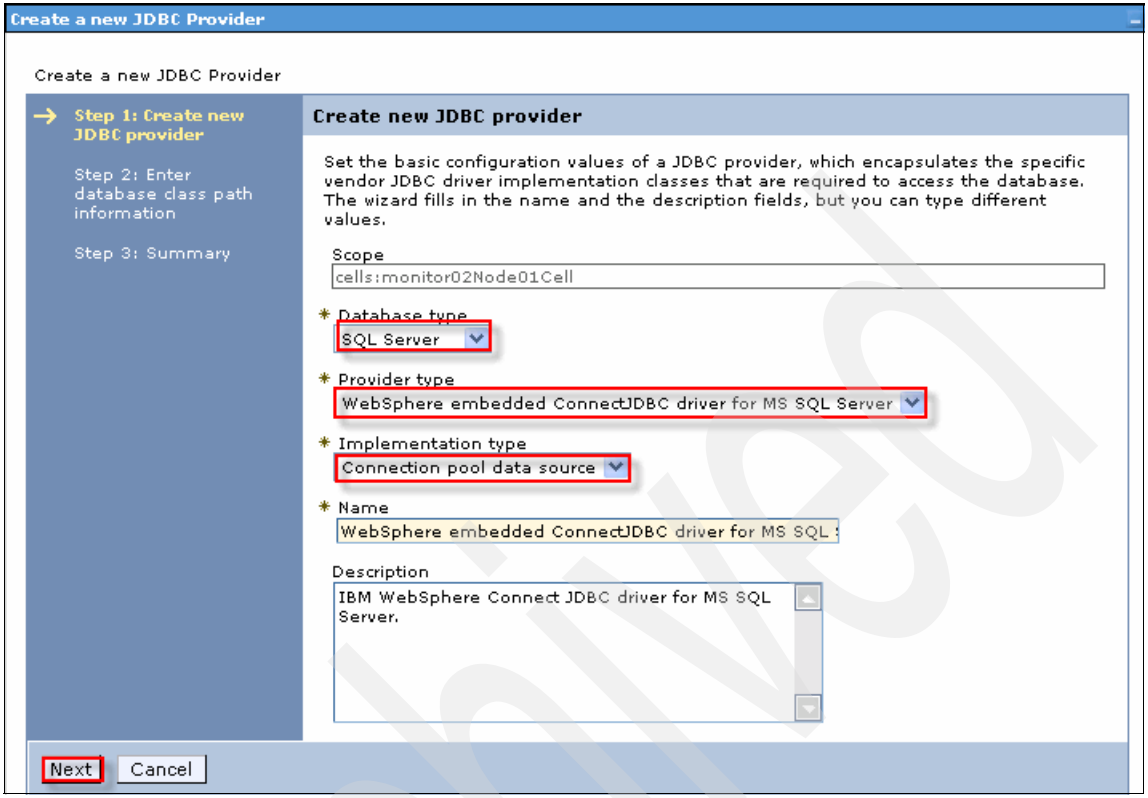


Figure 7-14 Creating a new JDBC Provider

Archived

d. Verify the summary and click **Finish** (Figure 7-15).

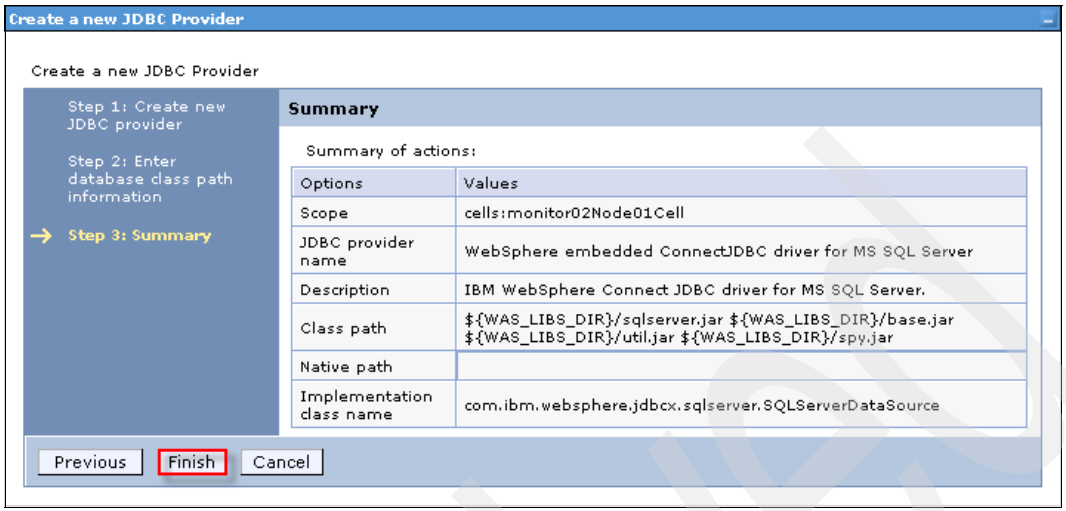


Figure 7-15 Summary of creating a new JDBC Provider

e. Click the **Save** directly to the master configuration link.

2. Create a data source for connecting to the process engine:

   a. In the WebSphere Administrative Console, expand **Resources** → **JDBC** and click **Data sources**.

   b. In the Create a data source panel, for Scope, select **Node=<host name>Node01**. In our example, we use **Node=monitor02Node01**. Click **New**.

Archive

c. In the Enter basic data source information panel (Figure 7-16):

   i. For Data source name, type `PEDataSource`.

   ii. For JNDI name, type `jdbc/com/ibm/P8/BPM/PEDataSource`.

   iii. For Component-managed authentication alias, select **monitor02Node01/PEDBUser**.

   iv. Click **Next**.

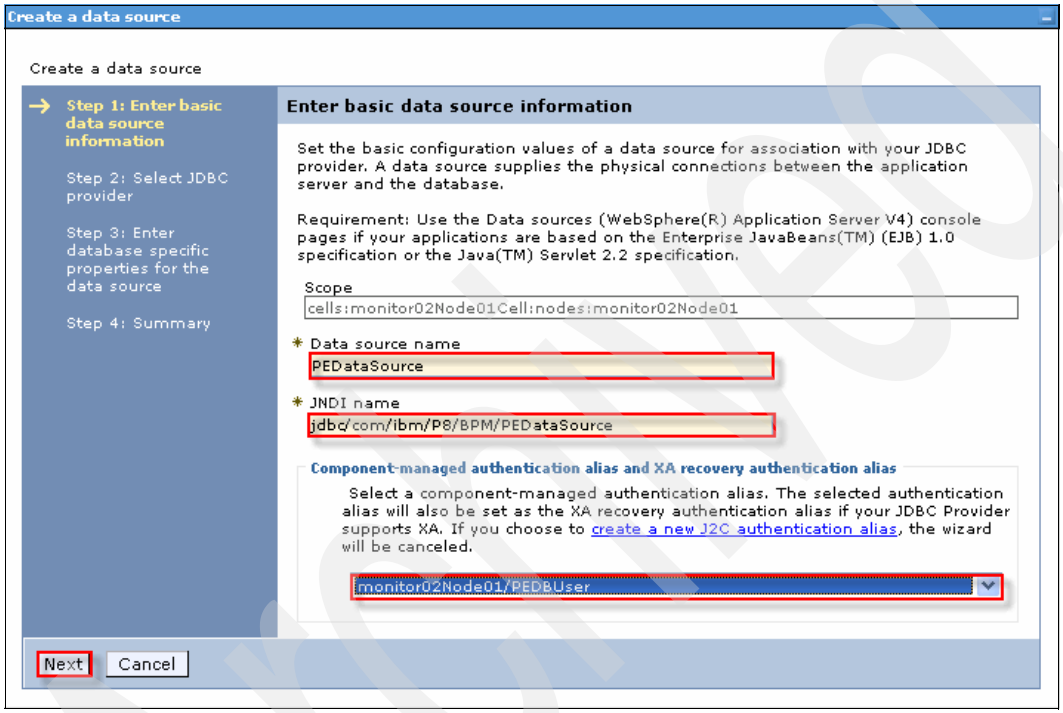


Figure 7-16 Creating a data source

d. Click **Select an existing JDBC provider**.

Archived

e. In the Select JDBC provider panel (Figure 7-17), select the JDBC Provider for connecting to the process engine database. In this example, we select **WebSphere embedded ConnectJDBC driver for MS SQL Server**. Click **Next**.

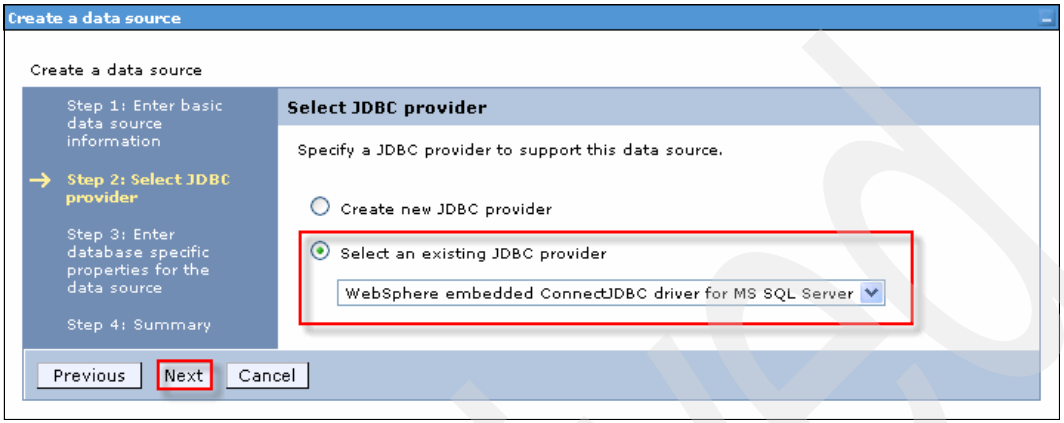


Figure 7-17 Selecting the JDBC provider

Archived

f. In the Enter database specific properties for the data source panel (Figure 7-18):

   i. For the Database name (process engine database name), type `VWdb`.

   ii. For the Server name (process engine database server name), type `hqdemo1`.

   iii. For Port number (process engine database port number), type `1433`.

   iv. Clear the **Use this data source in container managed persistence (CMP)** check box.

   v. Click **Next**.

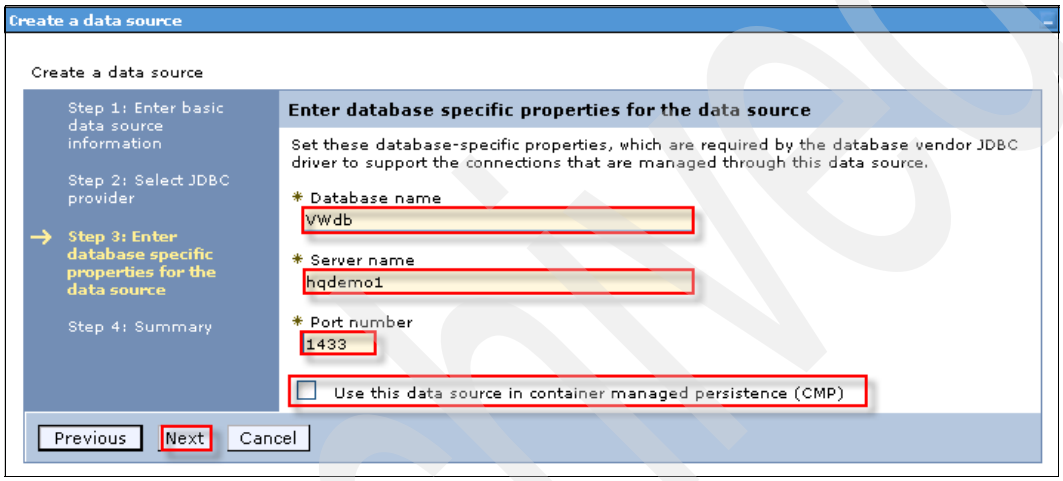


Figure 7-18 Entering the database-specific properties for the data source

URL property for the Oracle database: For the Oracle database, you specify a single URL property. The Oracle JDBC Driver data source requires the following URL entry, in place of the process engine database name and port:

```
jdbc:oracle:thin:@<host name>:<port>:<instancename>
```

Archived

g. In the Summary panel (Figure 7-19), verify the information and click **Finish**.

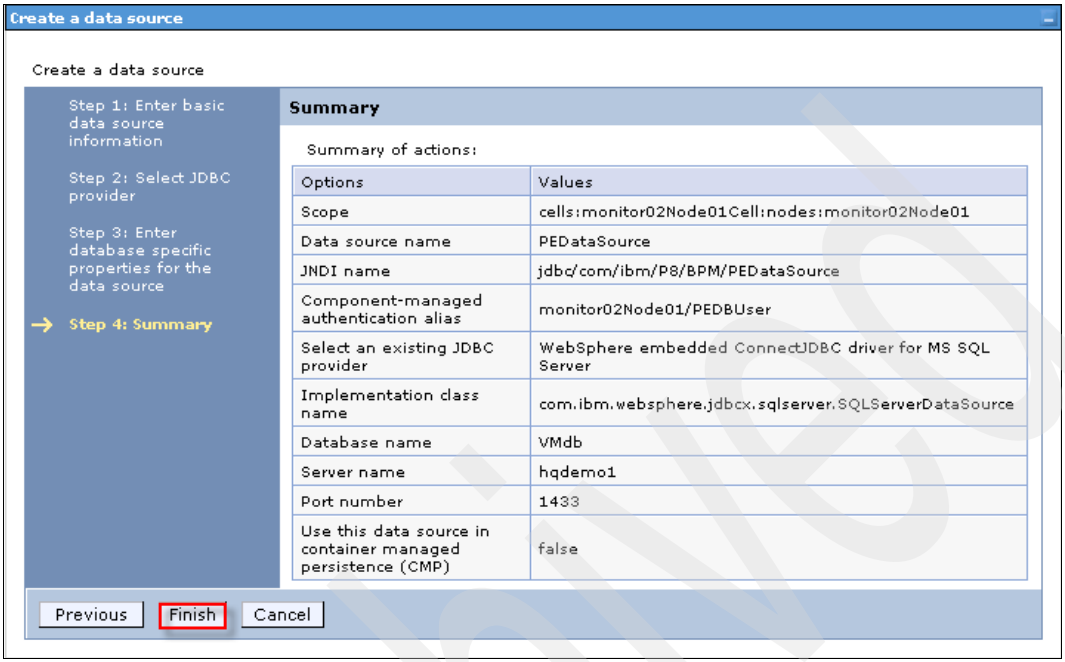


Figure 7-19 Summary of creating a data source

h. Click the **Save** directly to the master configuration link.

Archive

i. In the Data sources panel (Figure 7-20), select the **PEDataSource** check box and click **Test Connection** to verify the connectivity.

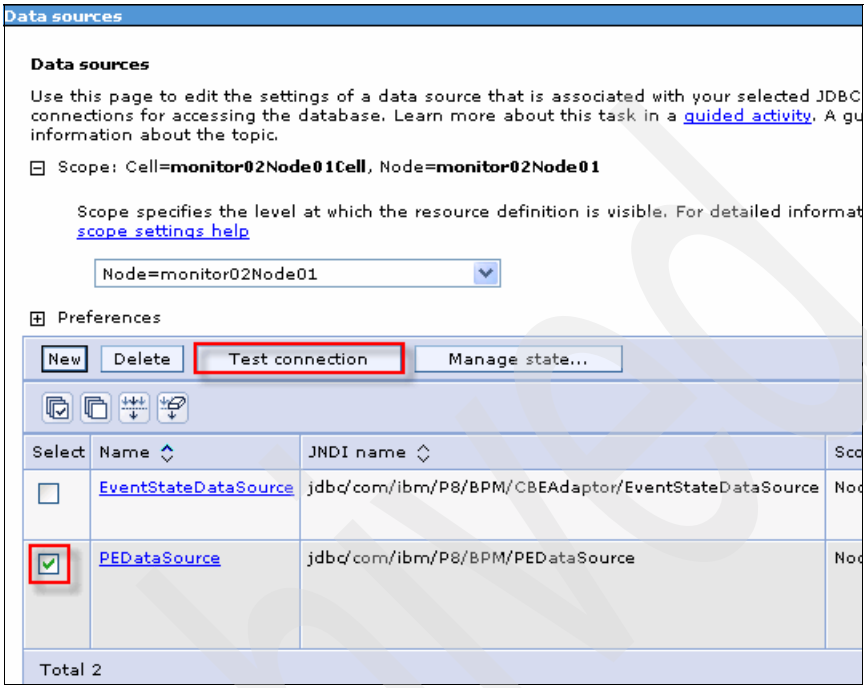


Figure 7-20 Testing the connection

If the test connection is successful, you see the test connection success message (Figure 7-21).

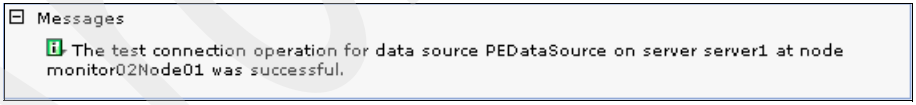


Figure 7-21 Successful test connection message

3. Create a JDBC provider to connect to the BPMSTATE database:

   a. In the WebSphere Administrative Console, expand **Resources** → **JDBC JDBC** and click **Providers**.

   b. In the Create a new JDBC Providers panel, select **Node=<host name>Node01** for the scope. In our example, we use **Node=monitor02Node01**. Click **New**.

Archived

c. In the next panel:

   i. For Database type, select **DB2**.
   ii. For Provider type, select **DB2 Universal JDBC Driver Provider (XA)**.
   iii. Select **XA data source** for the implementation Type.
   iv. Click **Next**.

d. Click **Next** to accept the default classpath information.

e. Verify the summary and click **Finish.**

f. Click the **Save** directly to the master configuration link.

Classpath and Native library path: The Classpath and Native library path of DB2 Universal JDBC Driver Provider (XA) refer to the WebSphere variables $DB2UNIVERSAL_JDBC_DRIVER_PATH and $DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH. Therefore you must set the values of these variables to `<DB2 INSTALL ROOT>/java`.

4. Create a data source for connecting to the BPMSTATE database:

   a. In the WebSphere Administrative Console, select **Resources** → **JDBC** → **Data sources**.

   b. In the Create a data source panel, for Scope, select **Node=<host name>Node01**. In our example, we use **Node=monitor02Node01**. Click **New**.

   c. In the next panel:

      i. For Data source name, type `EventStateDataSource`.

      ii. For JNDI name, type `jdbc/com/ibm/P8/BPM/CBEAdaptor/EventStateDataSource`.

      iii. For Component-managed authentication alias, select **monitor02Node01/BPMSTATE_User**.

      iv. Click **Next**.

   d. Click **Select an existing JDBC provider**. Select **DB2 Universal JDBC Driver Provider (XA)**. Click **Next**.

   e. In the next panel:

      i. For the Database name (Common Base Event Adapter Database name), type `BPMState`.

      ii. For Driver type, use the default of `4`.

      iii. For the Server name (Common Base Event Adapter Database Server name), type `localhost`.

iv. For Port number (Common Base Event Adapter Database Port number), type 50000.

v. Clear the **Use this data source in container managed persistence (CMP)** check box.

vi. Click **Next**.

f. Verify the summary and click **Finish**.

g. Click the **Save** directly to the master configuration link.

h. Select the **EventStateDataSource** check box and click **Test Connection** to verify connectivity.

If the test connection succeeds, you see the test connection success message.

7.4.5 Installing the FileNet P8 BPM Common Base Event Adapter

To deploy the FileNet P8 BPM Common Base Event Adapter for WebSphere:

1. In the WebSphere Administrative Console menu, select **Applications** → **Enterprise Applications**.
2. In the Enterprise Applications panel (Figure 7-22), click **Install**.

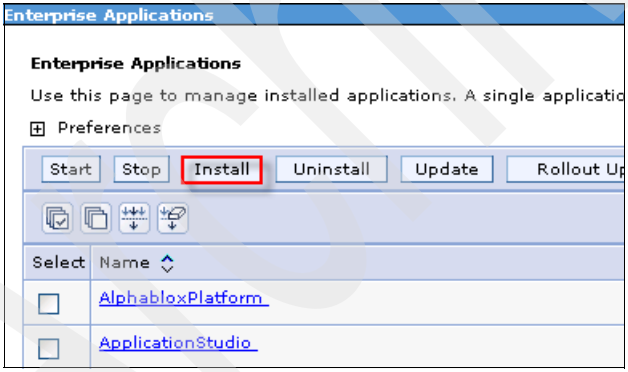


Figure 7-22 Enterprise Applications panel

3. In the Preparing for the application installation panel (Figure 7-23), for Local file system, click **Browse**. Select the **P8_BPM_CBEAdaptor.ear** file. Click **Next**.

> **P8_BPM_CBEAdaptor.ear file:** The P8_BPM_CBEAdaptor.ear file is supplied in FileNet P8 FixPack 4.0.2.

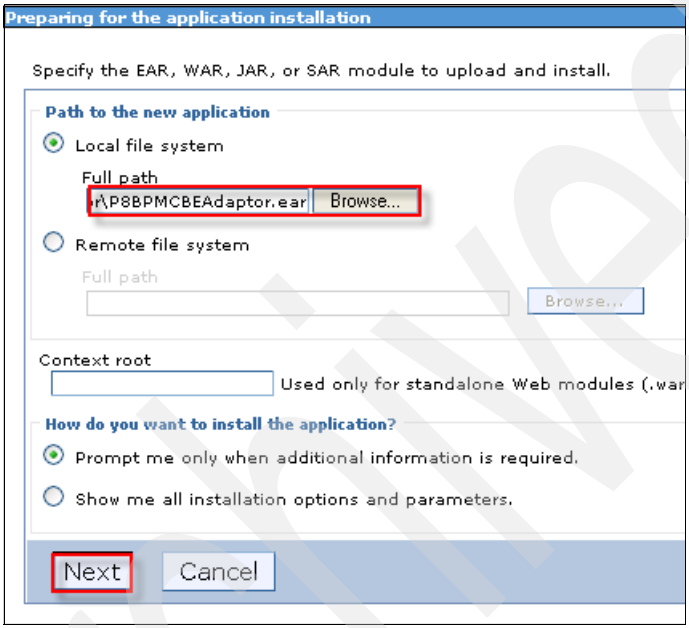


Figure 7-23 Preparing for the application installation

4. In the next few windows, click **Next**, click **Next** again, and then click **Finish** respectively.

Archived

5. When the Common Base Event Adapter installation is successful, you see the message `Application P8 BPM installed successfully` (Figure 7-24). Click the **Save** directly to the master configuration link.

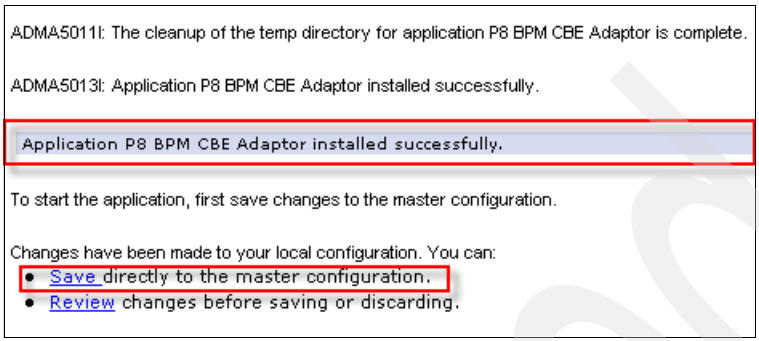


Figure 7-24 FileNet P8 BPM Common Base Event Adapter Installation Result

7.4.6 Setting optional configuration properties

To set optional configuration properties to fine-tune the behavior of the FileNet BPM Common Base Event Adapter for WebSphere:

1. In the WebSphere Administrative Console, expand **Servers** and click **Application servers**.
2. In the Application servers panel (Figure 7-25), click **server1**.

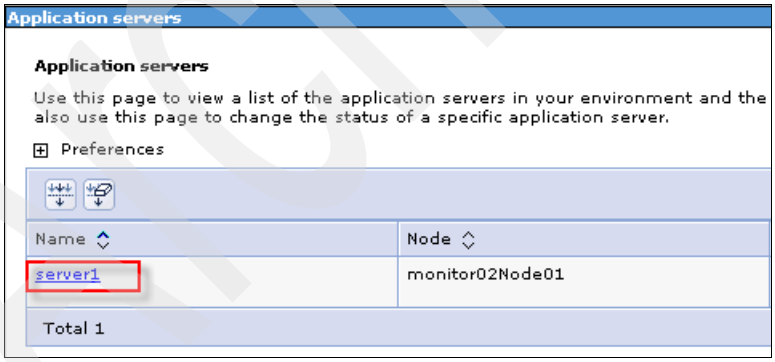


Figure 7-25 Application servers

3. Under the Server Infrastructure section (Figure 7-26), expand **Java and Process Management** and click **Process Definition**.

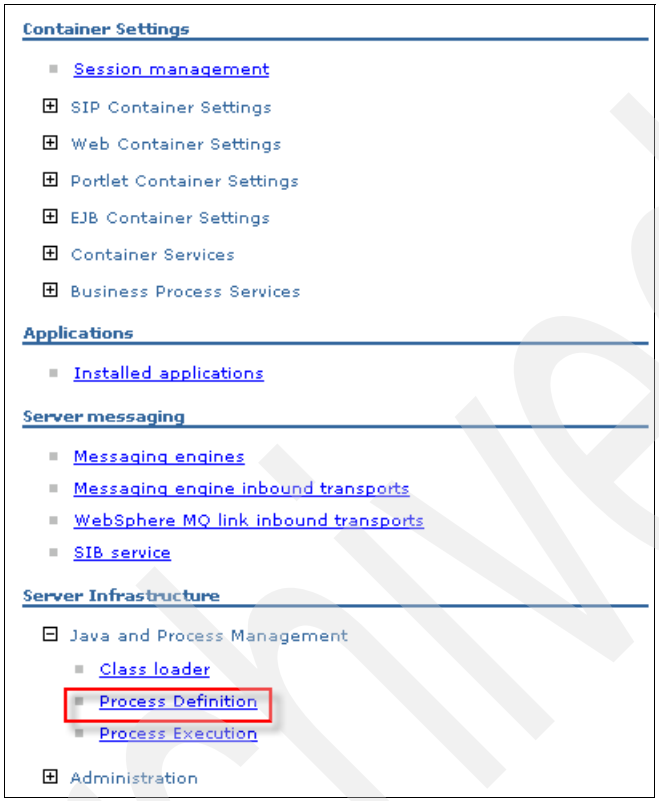


Figure 7-26 Process Definition

4. Under Additional Properties (Figure 7-27), click **Java Virtual Machine**.

Figure 7-27 Java Virtual Machines

Archived

5. Click **Custom Properties** (Figure 7-28).

Figure 7-28 Custom Properties

6. Click **New** to create any of the optional properties listed in Table 7-2.

Table 7-2 Optional properties

Name	Description
com.ibm.p8.bpm.cbe.adaptor.maxEventsPerSecond	This property restricts the rate at which the Common Base Event Adapter sends events to the Monitor server. If this property is not defined, the Common Base Event Adapter sends events at a maximum rate of 100 per second. The Common Base Event Adapter processes events much faster than the Monitor server consumes them. If this occurs, the Monitor server's JMS queue fills up, and the Common Base Event Adapter gets exceptions when it attempts to send new events.
com.ibm.p8.bpm.cbe.adaptor.statsInterval	This is the interval, in seconds, at which statistics are written to the log file. If the property is not defined, statistics are written every hour.

7. Click the **Save** directly to the master configuration link.
8. If you created any optional Java properties, restart the server before proceeding.

Archive

7.4.7 Starting the FileNet P8 BPM Common Base Event Adapter for WebSphere

To start the FileNet P8 BPM Common Base Event Adapter for WebSphere:

1. In the WebSphere Administrative Console, expand **Applications** and click **Enter Applications**.
2. In the Enterprise Applications panel (Figure 7-29), select the **P8_BPM_CBE_Adaptor** check box. Click **Start**.

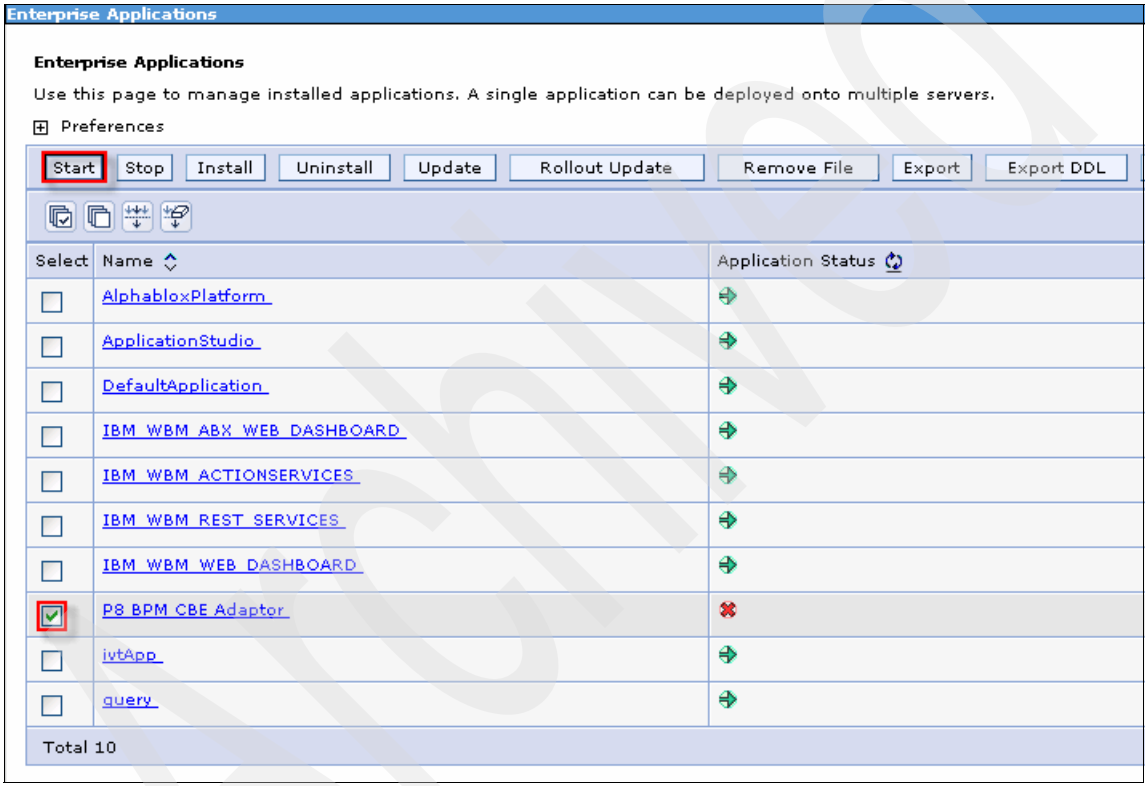


Figure 7-29 Starting the P8_BPM_CBE_Adaptor application

You see that the Application Status of P8_BPM_CBE_Adaptor changes to *Started*.

Archived

Logged messages: Messages logged by the FileNet P8 BPM Common Base Event Adapter for WebSphere are written to the following files:

- ${SERVER_LOG_ROOT}/SystemOut.log

  The System.out log is used to monitor the health of the running application server.

- ${SERVER_LOG_ROOT}/SystemErr.log

  The System.err log contains exception stack trace information useful for performing problem analysis.

7.5 FileNet P8 BPM Common Base Event events

In this section, Table 7-3 through Table 7-9 on page 383 describe the Common Base Event events that are emitted by FileNet P8 BPM when integrating with WebSphere Business Monitor. The Common Base Event specification is a standard XML-based format for business events, system events, and performance information. The FileNet P8 BPM Common Base Event Adapter for WebSphere Business Monitor converts FileNet P8 BPM events to the standard format required by WebSphere Business Monitor. Then it transmits the events via the CEI to the WebSphere Business Monitor server.

Table 7-3 Header information (predefined data)

Common Base Event attribute	Description	Process engine BPM attribute
version		Set to 1.0.1
creationTime	The time at which the event is created, UTC time stamp	F_TimeStamp
extensionName	Set to the BPM event name	F_EventType (in text format)
sequenceNumber		F_SeqNumber (monotonically increasing but not sequential)

Table 7-4 Correlation data (property data)

Common Base Event attribute	Description	Process engine BPM attribute
processId	Set to the process instance ID	F_WorkFlowNumber
workItemId	Set to the work item instance ID	F_WobNum
parentId	Set to the instance ID of the parent work item	F_ParentWobNum

Table 7-5 Common extended data

Common Base Event attribute	Description	Process engine BPM attribute
processName	Process definition name	Derived from F_WorkClassId
processVersion	Process version	F_WCRevision
processStartTime	Process start time	F_StartTime
logName	Event log view name	
isolatedRegion	Isolated region	
username	User that executed the action	F_UserId
businessData	Set of user-defined fields	

Table 7-6 Process events

Event	Description
P8.BPM.Process.Start	Event is emitted when the process starts.
P8.BPM.Process.End	Event is emitted when the process ends.

Table 7-7 Work item events

Event	Description
P8.BPM.WorkItem.Start	Event is emitted when a work item starts.
P8.BPM.WorkItem.End	Event is emitted when a work item ends.

Archive

Table 7-8 Activity events

Common Base Event attribute	Description	Process engine BPM attribute
activityId	Activity ID	<F_InstrSheetId>+<F_WorkOrderId>
map	Map (same as scope) name	Derived from F_InstrSheetId
activity	Step name	Derived from F_WorkOrderId
queue	Queue used by this activity	Derived from F_WPClassId
operation	Operation in the queue used by this activity	Derived from F_OperationId
boundUser	User to which the work item is assigned	F_BoundUserId
enqueueTime	Time the work item was placed in the current queue	F_EnqueueTime

Table 7-9 Activity events

Event	Description
P8.BPM.Activity.Start	Event is emitted when an activity starts. In the case of activity reassignment or delegation, multiple P8.BPM.Activity.Start events may be emitted for the same activity. This is as designed and is necessary since the process engine generates multiple 352 events for this case. Monitor models must be able to handle multiple P8.BPM.Activity.Start events.
P8.BPM.WorkItem.End	Event is emitted when a work item ends.
P8.BPM.Activity.Locked	Event is emitted when an activity is locked for processing.
P8.BPM.Activity.Unlocked	Event is emitted with an activity is unlocked.
P8.BPM.Activity.Failure	Event is emitted when an activity fails.

User-defined fields

In FileNet P8 BPM, fields can be exposed to an event log to capture the values of user-defined fields within a process. These user-defined fields are included in all of the FileNet P8 BPM Common Base Event events. The fields are stored in an extended data element named *businessData*. The user can access the user-defined fields via XPath expressions when building a monitor model in the Monitor Model editor.

Archive

7.6 FileNet P8 BPM Monitor Model for WebSphere Business Monitor

The FileNet P8 BPM software includes a basic monitor model so that WebSphere Business Monitor can monitor FileNet P8 BPM processes. A WebSphere developer uses the WebSphere Business Monitor Toolkit to augment the FileNet P8 BPM Monitor Model for WebSphere Business Monitor and add user-defined fields to monitor.

7.6.1 Monitor model

The monitor model provided for FileNet P8 BPM monitors active processes and active work items. It monitors the following items:

- Process count (by their workflow definition)
- Work item counts (by their steps, users and queues)
- Processing time
- Idle time
- Total processing time

You can add user-defined fields to extend the monitoring. The monitor model provided by FileNet P8 BPM has two types of context:

- *Process level context* defines how to monitor process-level events.
- *Work item level context* defines how to monitor work item-level events.

The monitor model always uses the process monitor context at the root level. It uses the work item monitor context at the children context level.

7.6.2 Process monitor context

The process monitor context contains the following elements

- Key

  A *key* is information that characterizes and identifies the real-world entity being tracked by the monitoring context. One or more metrics defined for a monitoring context must be declared key components. The set of these metrics, together with the key components of any ancestor monitoring contexts (parent, grandparent, and so on), defines a monitoring context's key.

  The process key, set to the workflow number of each process instance, is the key for process monitor context. It is the correlation data identifying the unique context instance.

Archived

- Inbound events

  A monitoring-enabled application generates a series of events. To indicate the events of interest to the monitoring context, define inbound events in the monitor model editor. These are subscription points or entry points for inbound events in a monitoring context. WebSphere Business Monitor subscribes to the events specified and delivers them to all event entry points with a matching event subscription.

  Configure the event filter so that only processes that start after the specified time will be considered by the monitor model. The inbound events of process monitor context are composed of a process start event and a process end event.

- Stopwatch

  KPIs often depend on elapsed time or on the number of occurrences of some situation or event. Define stopwatches and counters, which are specialized metrics, in the monitor model editor to track elapsed time or to count such occurrences.

  The *processing time stopwatch* measures how long the process has been in the system. It is started by the process start event and stopped by the process end event.

- Triggers

  A *trigger* is a mechanism that detects an occurrence and causes additional processing in the response. For example, defining a trigger that updates a metric, incrementing a counter or halting a stopwatch each time a task ends. A trigger can also cause a map to populate an outbound event.

- Metrics

  A *metric* is a holder for information, usually a business performance measurement, in a monitoring context. 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. Examples of metrics are the working duration of a process, the name of the user assigned to a task, a supplier's response time, and the cost of the risk assessment step in an insurance process.

Archived

7.6.3 Work item monitor context

The work item monitor context is created as a child of the process monitor context. It contains the following elements:

- Key

  A *key* is information that characterizes and identifies the real-world entity being tracked by the monitoring context. One or more metrics defined for a monitoring context must be declared key components. The set of these metrics, together with the key components of any ancestor monitoring contexts (parent, grandparent, and so on), defines a monitoring context's key.

  An activity instance ID is the key for a work item monitor context. This key and the process ID are the correlation data that identify the unique context instance. The activity instance ID denotes a specific work item.

- Inbound events

  The inbound events of the work item monitor context are composed of a work item start event, a work item end event, an activity start event, an activity locked event, an activity unlocked event, an activity end event, and an activity failure event.

- Stopwatch

  Processing time stopwatch measures how long the work item has been locked in the system. Idle time stopwatch measures how long the work item has been in the system and not in the process (locked) state.

- Triggers

  The triggers of the work item monitor context are composed of an activity locked trigger, an activity unlocked trigger, an activity end trigger, an activity failure trigger, a work item end trigger, and an evaluate stopwatch.

- Metrics

  The metrics of the work item monitor context are composed of region, process name, map, step, queue, operation, processing time, idle time, total processing time, user, log name, time stamp, and sequence number. Their values are set by different events according to FileNet Process Engine semantics.

More information: For more information about the process monitor context and work item context, see the online FileNet documentation at the following Web address:

`http://www-1.ibm.com/support/docview.wss?rs=3278&uid=swg27010422`

Archived

7.6.4 Data mart model

The data mart model is the part of the monitor model that defines the cubes that are used for storing, retrieving, and analyzing the data gathered over time:

- Cubes

  A data mart model contains one cube for each monitoring context definition. Each cube tracks all the information that the monitoring context collects over time. For each activity you monitor, there is a cube for that activity monitoring context. The cubes are named as follows:

  - <process name> cube
  - <activity1 name> cube
  - <activity2 name> cube

- Cube for process

  - Cube name: BPM process MC cube
  - Dimensions: Process (levels: region, process name, process version)
  - Measures: Average processing time

- Cube for work item

  - Cube name: BPM work item MC cube
  - Dimensions: User (level: user)
  - Queue (levels: queue and operation)
  - Activity (levels: region, process name, map, and step)
  - Measures: Average processing time
  - Average idle time
  - Average total processing time

- Dimension

  Dimensions are data categories that are used to organize and select instances for reporting and analysis. The primitive monitor model has no predefined dimensions.

7.7 Implementing the FileNet P8 BPM processes

In this section, we provide instructions for implement a Contract Management Workflow by using FileNet P8 BPM.

For details about Contract Management Workflow, see 4.3, “Contract management process” on page 145.

Archived

7.7.1 Implementing the Contract Management Workflow

To complete the Contract Management Workflow in FileNet P8 BPM, we proceed through the following steps:

1. Creating the work queues and exposing the data fields
2. Creating the process

Creating the work queues and exposing the data fields

We create two work queues to holds all the contract case work items during run time and expose those workflow data fields in these queues. Two work queues, including Indexing and Review, are required for this workflow. To create the work queues and expose the data fields:

1. Launch the FileNet Workplace. Open a browser and type the following URL:

   `http://[FileNet P8 BPM host]:9080/Workplace`

2. Log in as `administrator`.
3. As shown in Figure 7-30, in the left pane, select **Admin** and in the right pane, click **Process Configuration Console**.

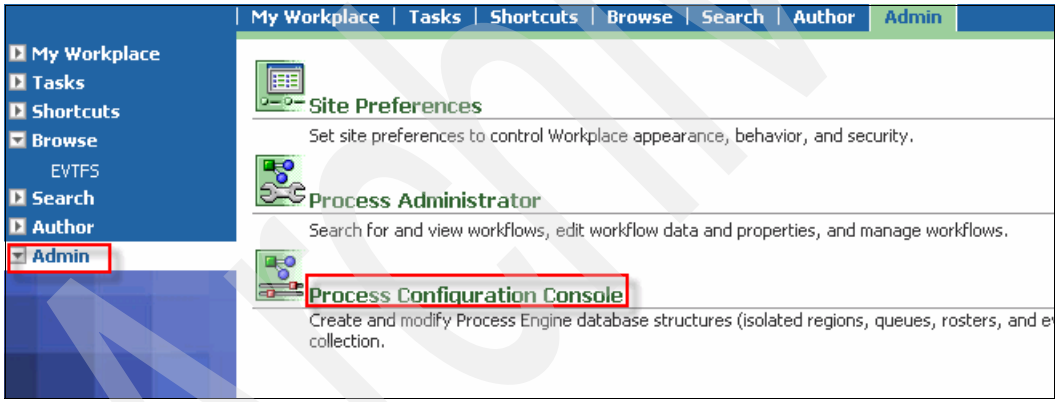


Figure 7-30 Accessing the Process Configuration Console

Archive

4. In the Process Configuration Console (Figure 7-31), right-click the **region name** and select **Connect**. Isolated region is the name of the BPM database.

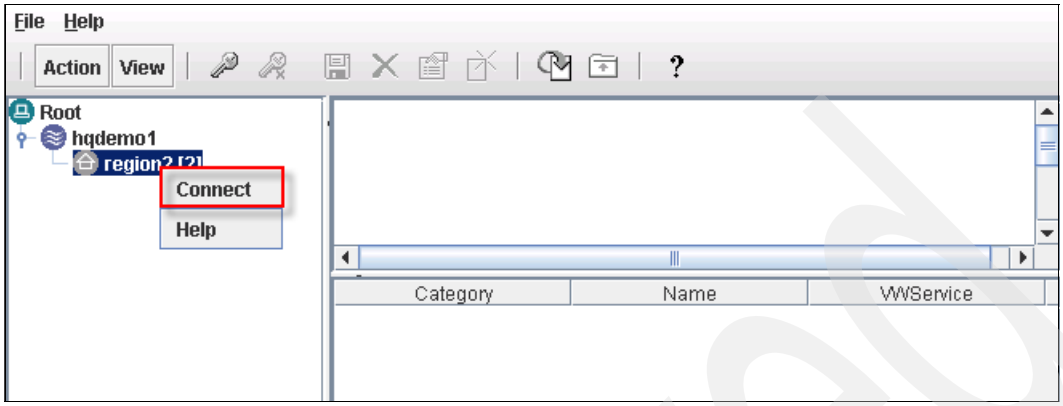


Figure 7-31 Connecting to the region

5. As shown in Figure 7-32, right-click **Work Queue** and select **New**.

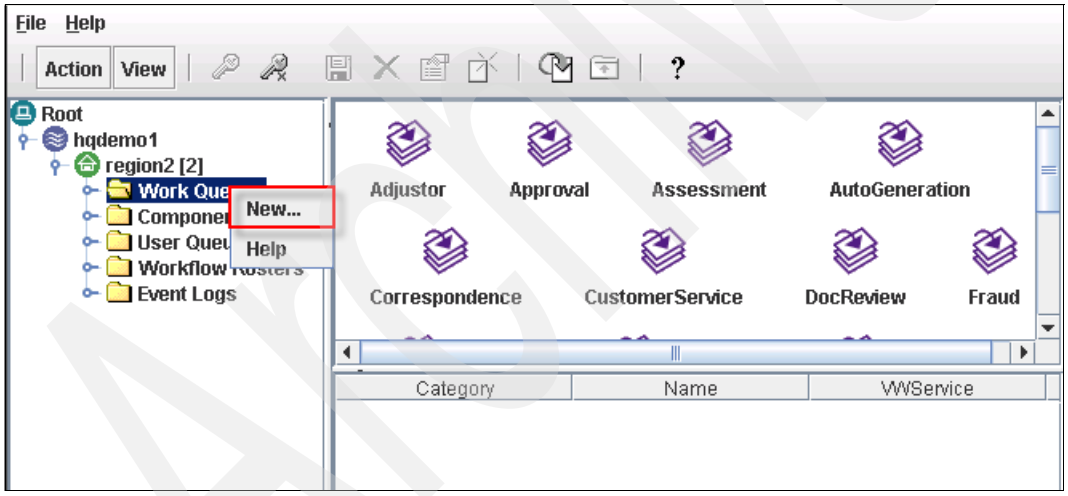


Figure 7-32 Creating the work queue

Archived

6. In the New Queue window (Figure 7-33), for Queue Name, type `Indexing`. Then click the **Create** button.

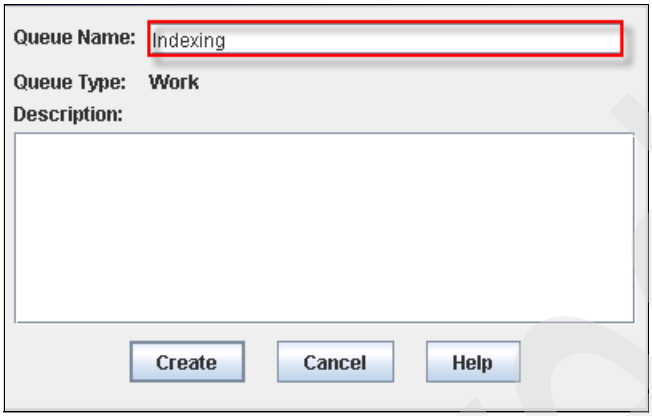


Figure 7-33 Create Indexing queue

7. Repeat steps 5 and 6 to create a new work queue named `Review`.
8. Right-click the region name and select **Commit Changes** (Figure 7-34).

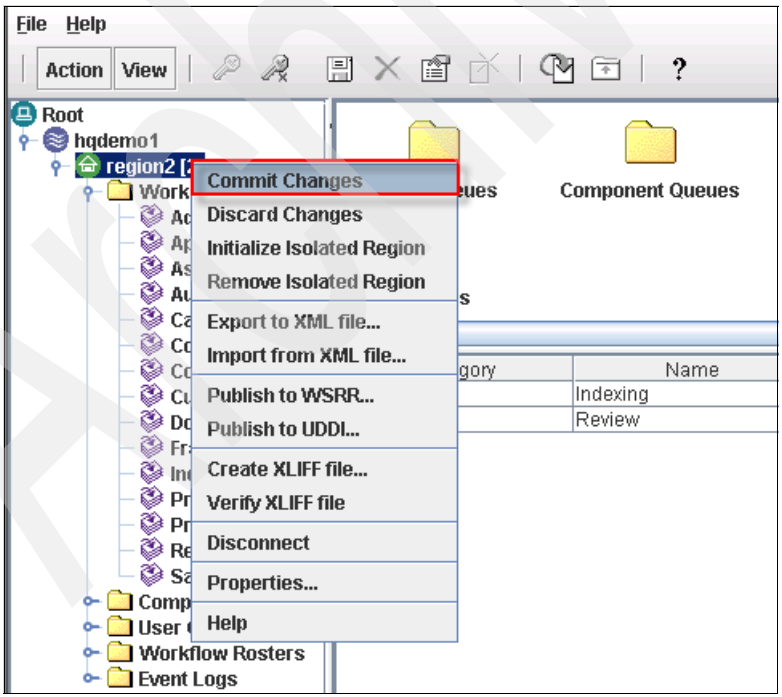


Figure 7-34 Selecting Commit Changes

Archived

9. In the Commit changes panel (Figure 7-35), click **Continue** and wait for the database table update to finish.

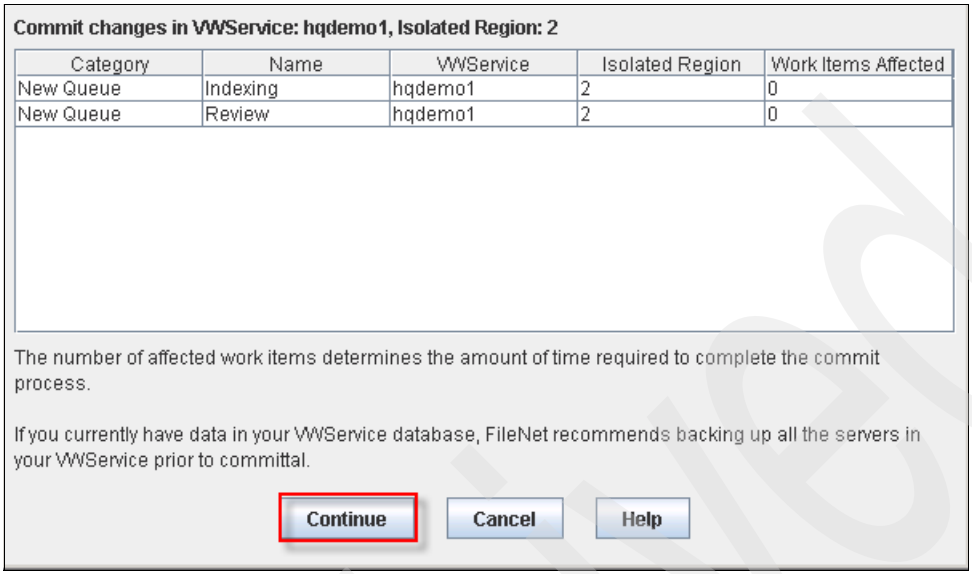


Figure 7-35 Commit Changes

10. Click **Close** to close the Java applet window. If the Java window is not closed, the Process Designer applet might not work properly.

    The Indexing and Review queues are created under Work Queues (Figure 7-36).

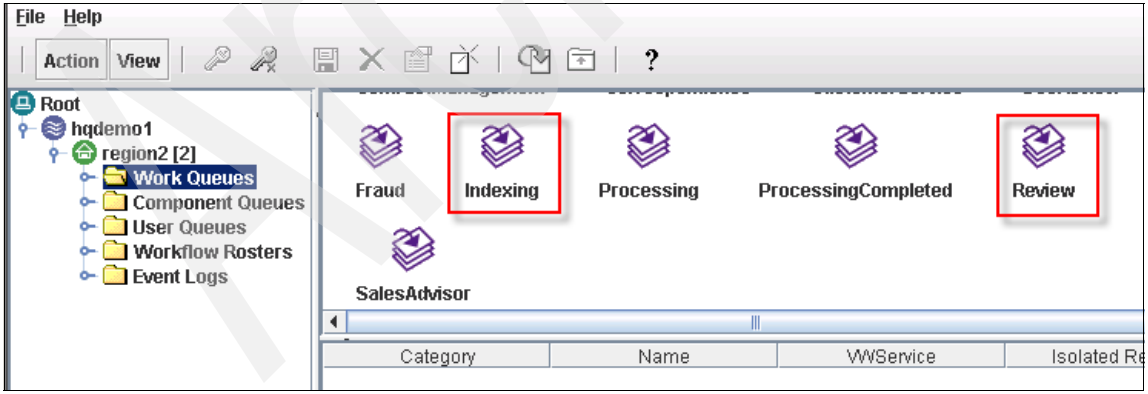


Figure 7-36 Indexing and Review queues

Archived

11. As shown in Figure 7-37, select **Event Logs**. Right-click **DefaultEventLog** and select **Properties**.

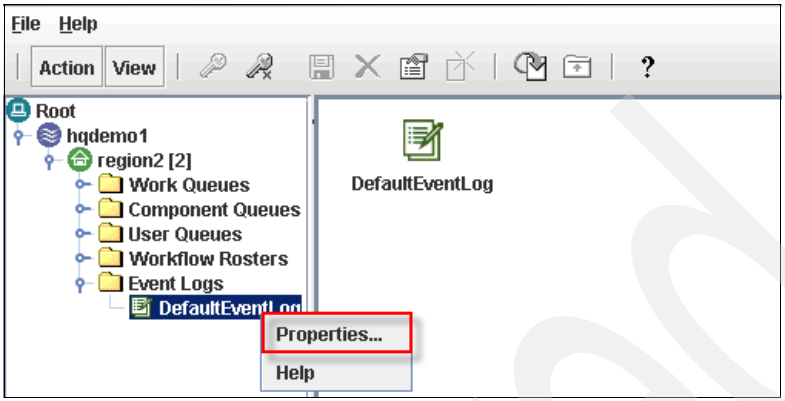


Figure 7-37 Selecting Event Logs

12. In the next window (Figure 7-38), click the **Data Fields** tab and enter three data fields: `CompanyName` (String), `ContractAmount` (Float), and `AccountNumber` (String). These fields are used in the Contract Management Workflow and ultimately monitored by using IBM WebSphere Business Monitor. Click **OK**.

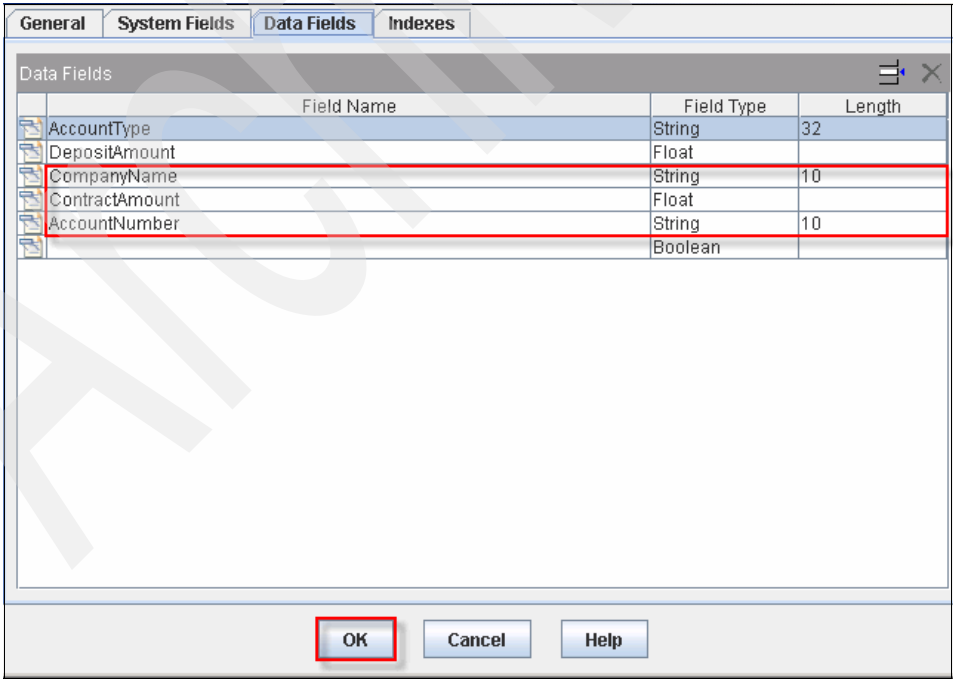


Figure 7-38 Event Logs: Data Fields

13. After updating the DefaultEventLog properties, select the region name, right-click and select **Commit Changes**.

Creating the process

The Process Configuration Console and Process Designer provide a way to implement the business process model. To load the Contract Management workflow into FileNet P8 BPM:

1. Launch the FileNet Workplace and log in as an administrator.
2. As shown in Figure 7-39, in the left pane, select **Author** → **Advanced Tools**. In the right pane, click **Process Designer** to start the Process Designer tool.

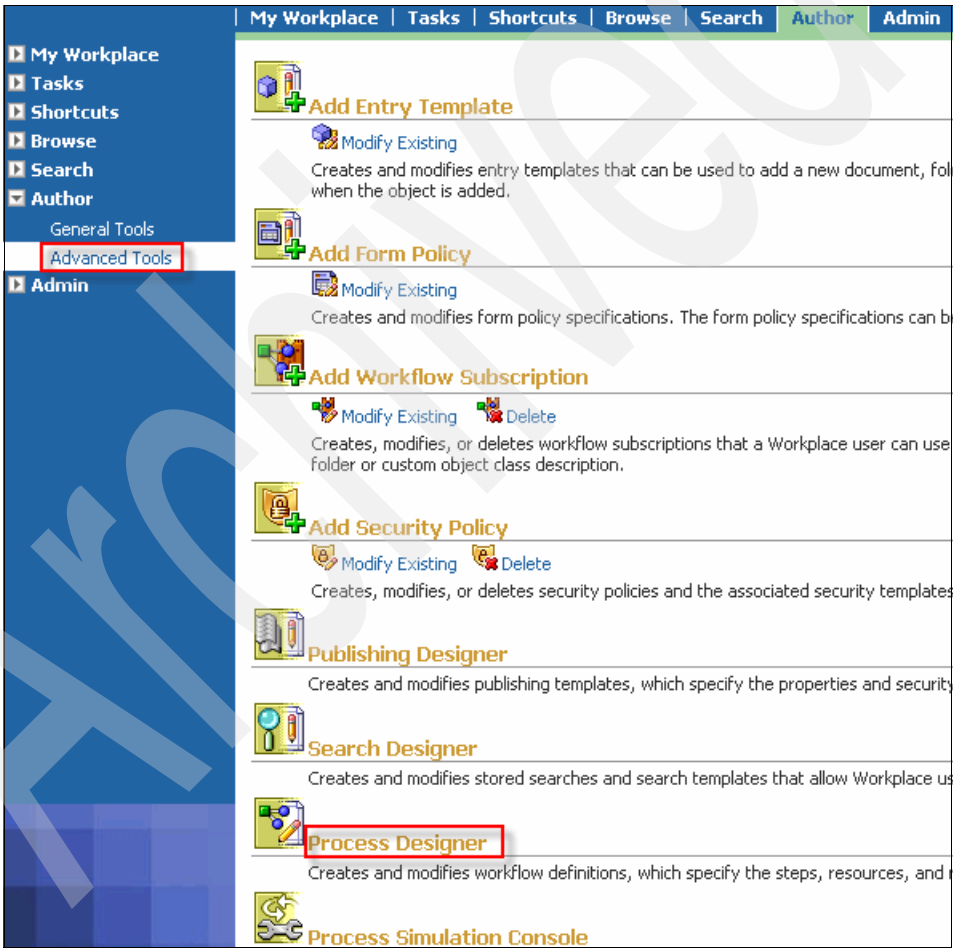


Figure 7-39 Starting the Process Designer tool

Archived

3. When the Process Designer applet is loaded, select **File** → **Open** (Figure 7-40).

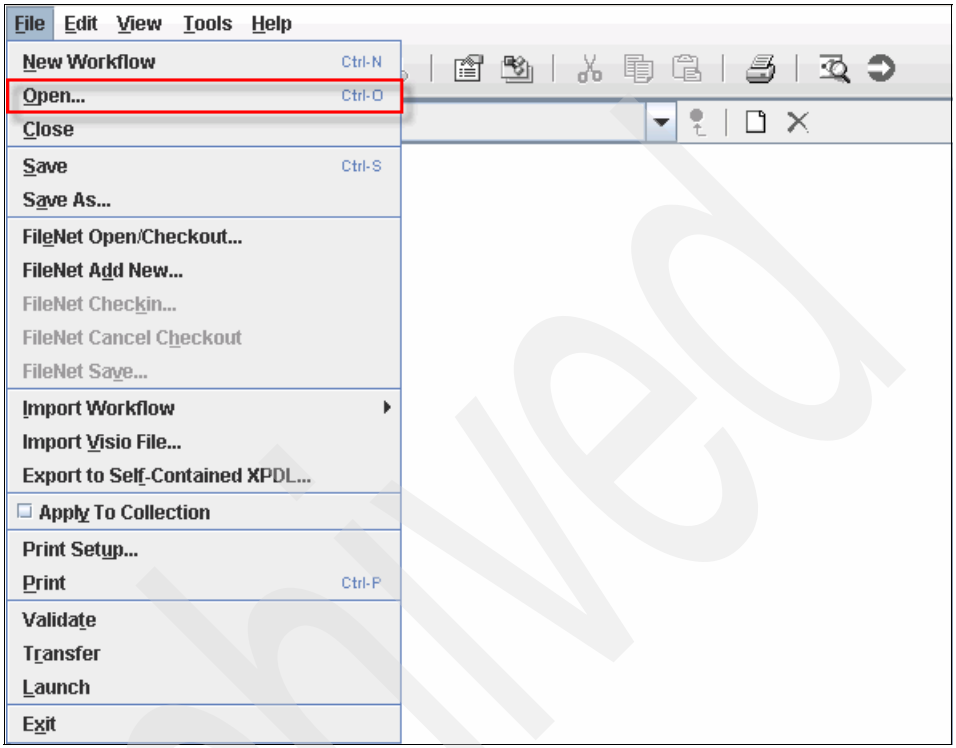


Figure 7-40 Opening a file in FileNet P8 Process Designer

Archived

4. In the Open panel (Figure 7-41), select **Contract Management Workflow v1.1.pep**, which is supplied with the additional materials for this book. See Appendix B, “Additional material” on page 801, for details. Click **Open**.

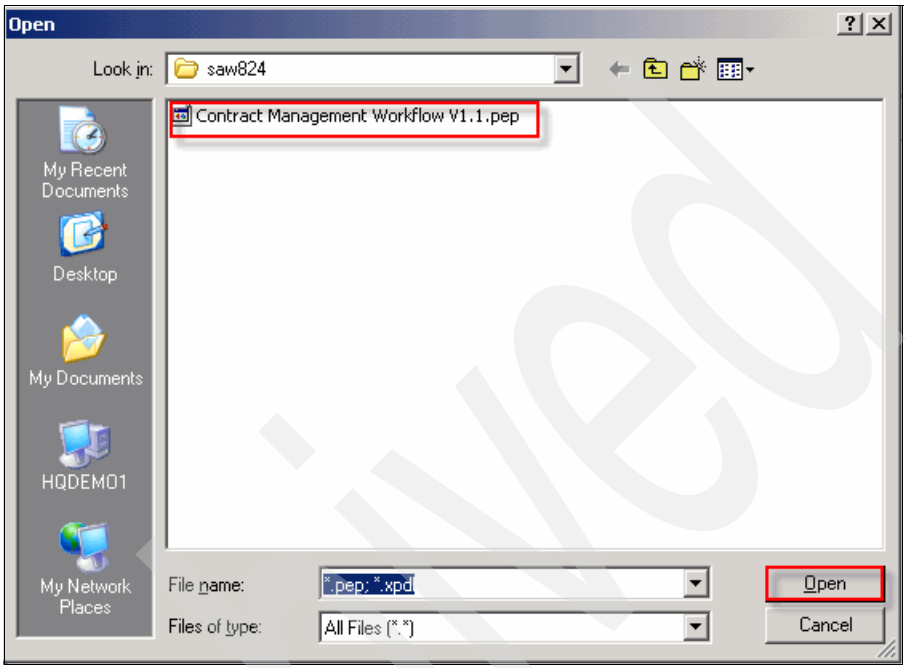


Figure 7-41 Open file window

Archived

5. In the Contract Management Workflow (Figure 7-42), from the menu bar, select **View** → **Workflow properties** to see the settings in the workflow.

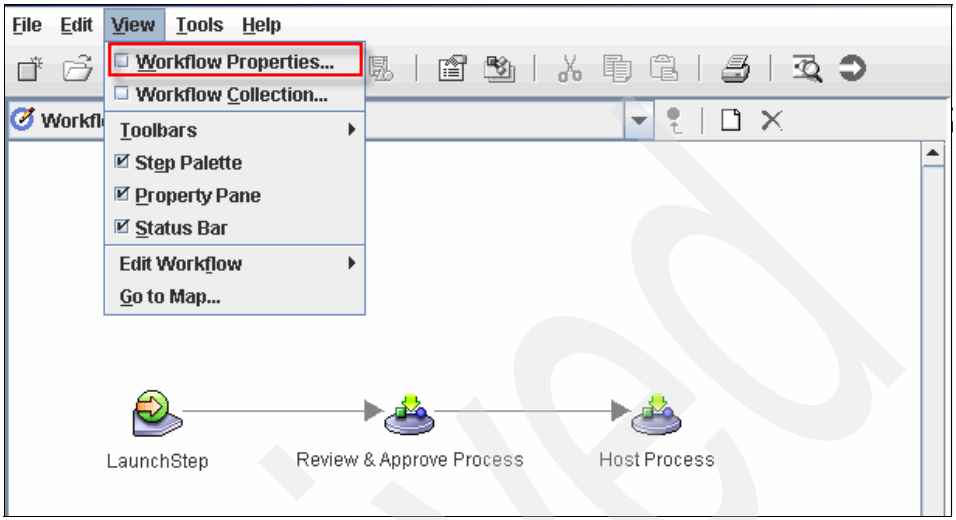


Figure 7-42 Selecting to view the Workflow Properties

6. The Workflow Properties panel (Figure 7-43) shows the properties of Contract Management Workflow, including the General, Advanced, Data Fields, Attachments, Workflow Groups, Maps, Milestones and Web Services tabs.

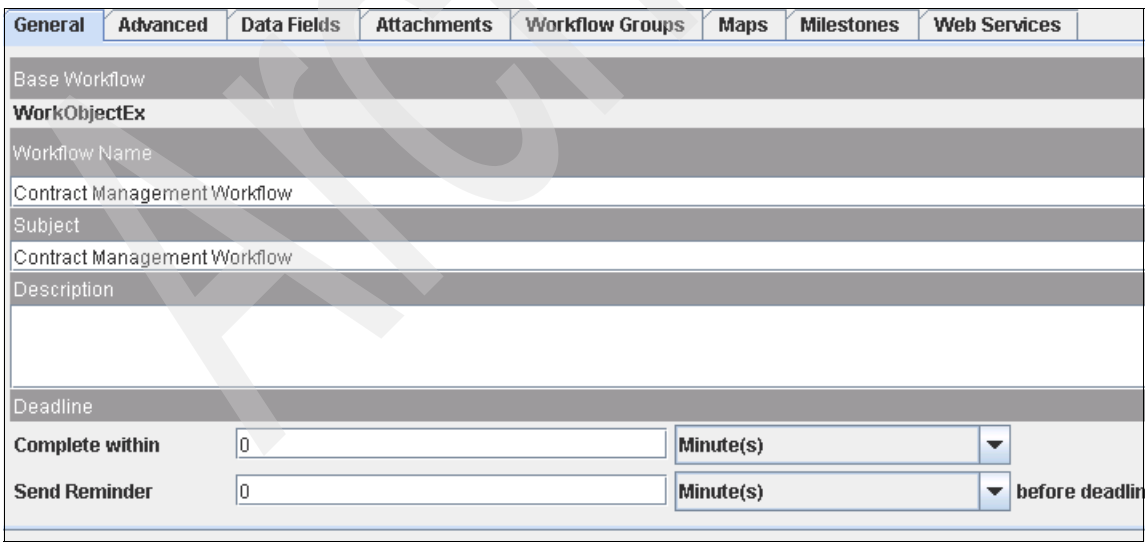
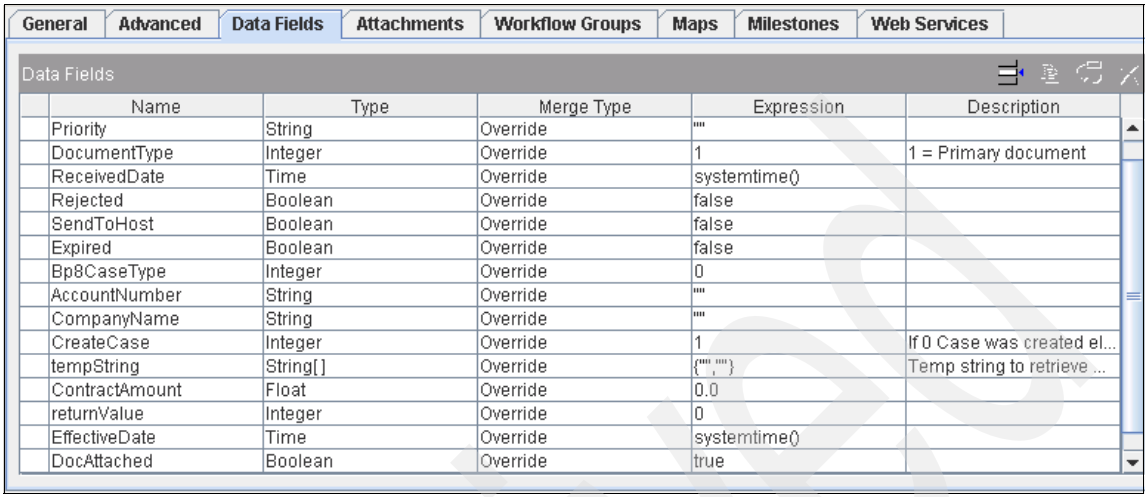


Figure 7-43 Workflow properties

Archive

a. Click the **Data Fields** tab (Figure 7-44) to see all data fields of Contract Management Workflow.



Name	Type	Merge Type	Expression	Description
Priority	String	Override	""	
DocumentType	Integer	Override	1	1 = Primary document
ReceivedDate	Time	Override	systemtime()	
Rejected	Boolean	Override	false	
SendToHost	Boolean	Override	false	
Expired	Boolean	Override	false	
Bp8CaseType	Integer	Override	0	
AccountNumber	String	Override	""	
CompanyName	String	Override	""	
CreateCase	Integer	Override	1	If 0 Case was created el...
tempString	String[]	Override	{"","",""}	Temp string to retrieve ...
ContractAmount	Float	Override	0.0	
returnValue	Integer	Override	0	
EffectiveDate	Time	Override	systemtime()	
DocAttached	Boolean	Override	true	

Figure 7-44 Data Fields page

b. To see the roles of the Contract Management Workflow, click the **Workflow Groups** tab (Figure 7-45 on page 398). Table 7-10 describes the roles involved in this process flow.

Table 7-10 Contract Management Workflow roles

Role name	Role description	Participants
Indexer	Collects the contract information and then creates and indexes the contract case	Group: Indexer
Reviewer	Reviews the contract case and then sends to approver to approve or reject back	Group: Reviewer
Approver	Finally approves or rejects the contract case	Group: Approver
Supervisor	Can be an indexer or reviewer; can search the finished case	Group: Supervisor
F_Trackers	Can track the workflow in graphical mode when it progresses	Administrator

Requirement: The group Indexer, Reviewer, Approver, and Supervisor must be in your current LDAP directory.

Archive

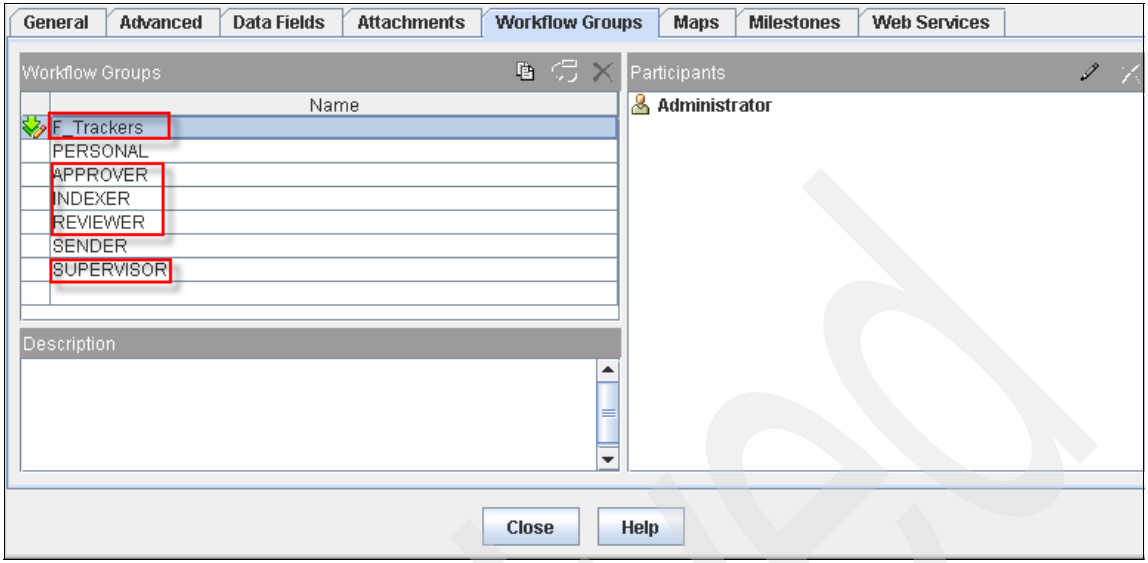


Figure 7-45 Workflow Groups

c. Click **Close** to close the Workflow Properties panel.

7. Investigate the Contract Management Workflow:

   a. Figure 7-46 shows the main workflow map, which is composed of two submaps, including Review_Approve Process and Host Process. In the main workflow map, double-click the **Review_Approve Process** icon.

Figure 7-46 Main workflow map

Archived

In the Review_Approve Process submap (Figure 7-47), we define three submaps for Indexer Process, Reviewer Process, and Approve Process respectively. The *20 Day Timer* is used to time the process.

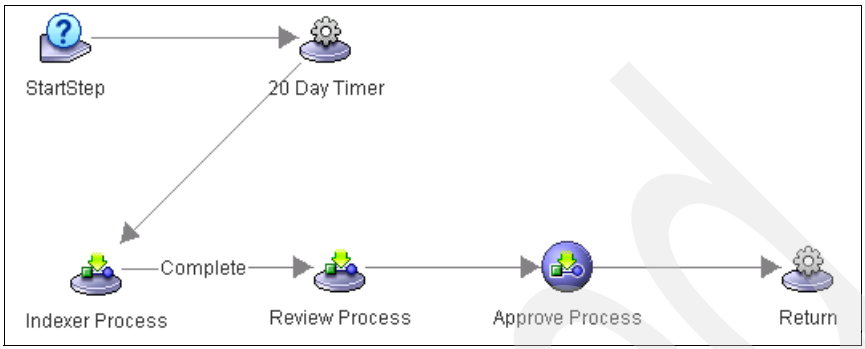


Figure 7-47 The Review and Approve submap

b. In the Review_Approve Process submap, double-click the **Indexer Process** icon to open the Indexer Process submap.

Figure 7-48 shows the submap definition of the Indexer Process, where the *indexing queue* step, as a general step, is configured to be in the indexing queue. Each user sees all the same contract cases in this step. The *pend* step is a participant step, configured to be in the indexer's personal queue, so that at this step, indexers can only see those cases.

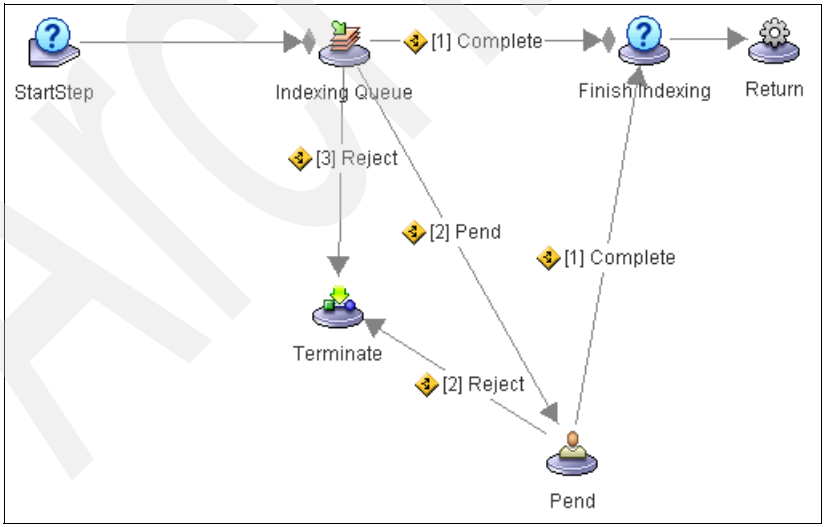


Figure 7-48 The Indexer Process submap

Archived

c. In the Review_Approve Process submap, double-click the **Review Process** icon to open the Review Process submap.

Figure 7-49 shows the definition of the Review Process submap. The *review queue* step is also put into the review queue, so that each user sees all of the same contract cases in this step. In addition, the *pend* step is a participant step, so that reviewers can only see those cases.

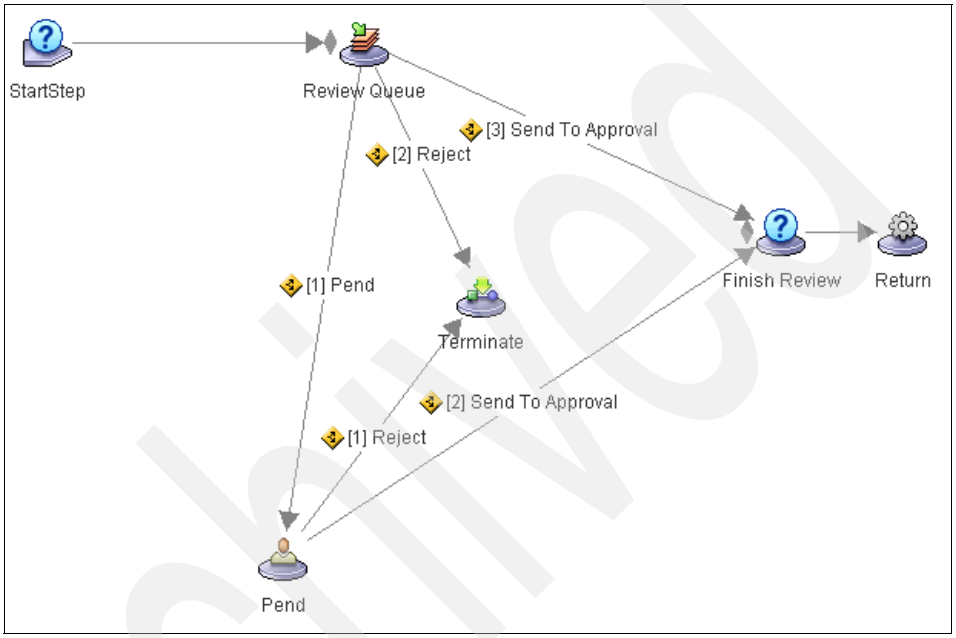


Figure 7-49 The Review submap

Archived

d. In the Review_Approve Process submap, double-click the **Approve Process** icon to open the Approve Process submap.

Figure 7-50 shows the definition of the Approve. The *approve queue* step is not in the queue, but rather it is a participant step. This means that the approver can only approve those contract cases sent by the reviewer.

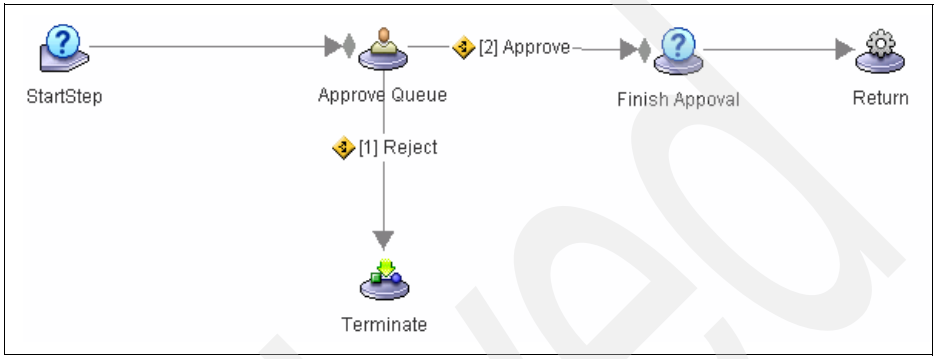


Figure 7-50 The Approve submap

8. In the main workflow map, double-click the **Host Process** icon to open the Host Process submap.

Figure 7-51 shows the definition of the Host Process, which stops all timers and finishes the process.

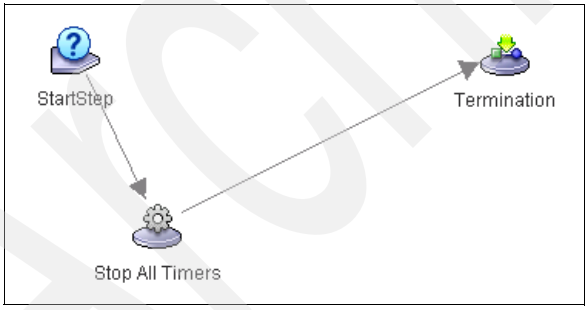


Figure 7-51 The Host Process submap

Archived

9. After investigating the Contract Management Workflow, transfer the workflow to the object store:

   a. Select **File** → **Transfer** (Figure 7-52).

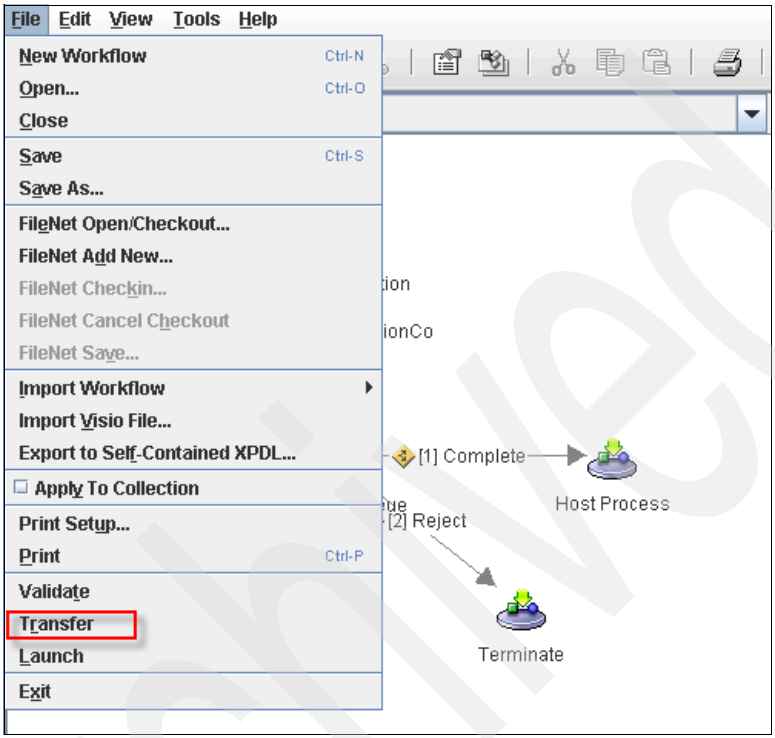


Figure 7-52 Selecting Transfer to transfer the workflow

   b. In the Save the workflow definition to the object store panel (Figure 7-53 on page 403):

      i. For Folder, click **Browse** to choose the object store (library) and folder to save the workflow definition XML file.

      ii. In the Select a Folder window, double-click **EVTFS** for the object stores and select **Workflows** for the folder. Then click the **Select** button.

      iii. Click **Next**.

Archived

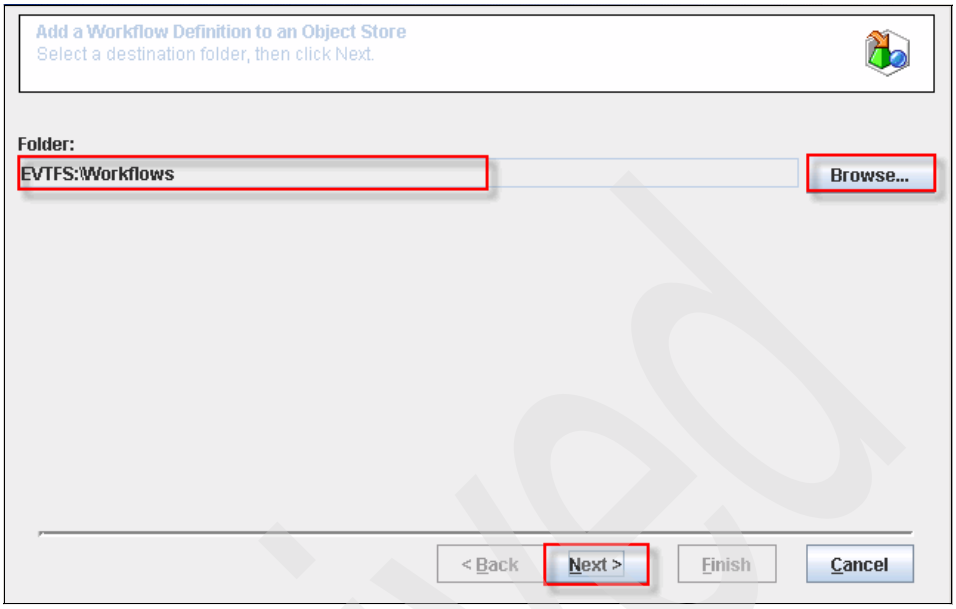


Figure 7-53 Adding a workflow definition to an object store

c. Under Properties (Figure 7-54), for Document Title, type `Contract Management Workflow` and click **Next**.

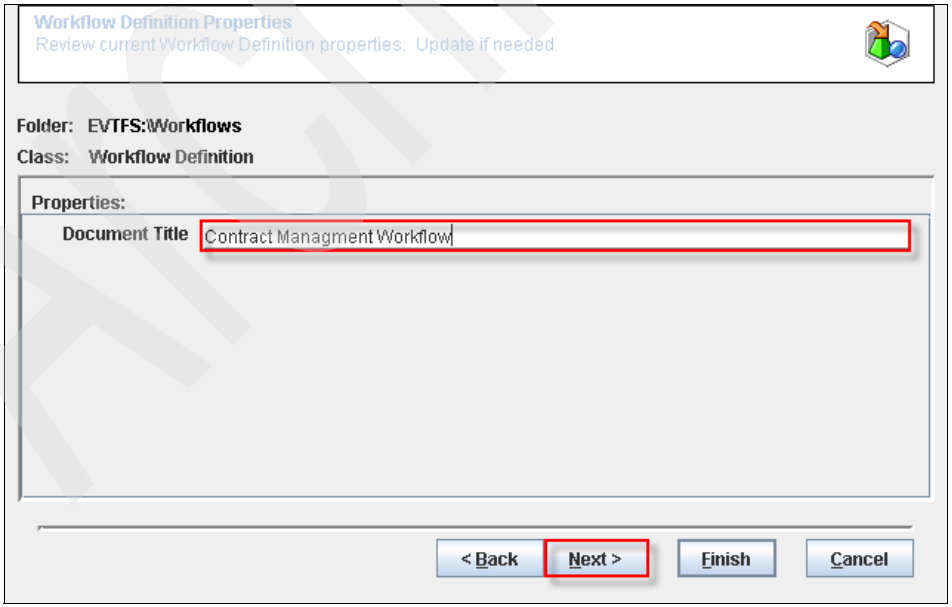


Figure 7-54 Workflow definition properties

Archived

d. In the Security Settings panel (Figure 7-55), click **Finish**.

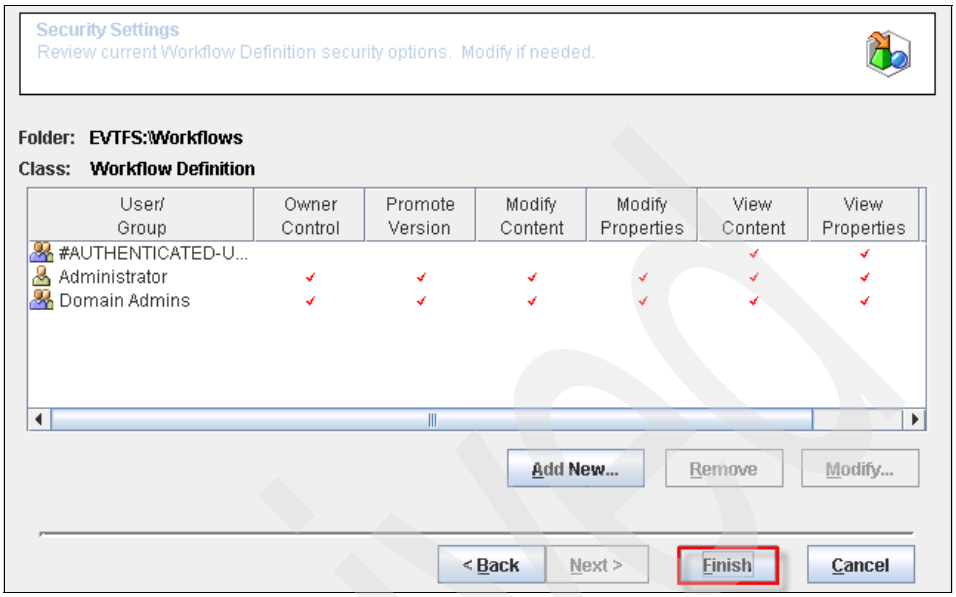


Figure 7-55 Security Settings panel

After the transfer is finished, a window opens that prompts that the transfer was successful. Now you can launch the workflow.

7.8 Running the Contract Management Workflow

The application is now ready to run via the FileNet Workplace.

User and roles

To run the Contract Management Workflow, we must have the users in the LDAP directory, which stores users of FileNet P8, as described in Table 7-11.

Table 7-11 Users and roles

Role	User	Password
Indexer	joe	password
Reviewer	mark	password
Approver	ana	password
Supervisor	sue	password

Business process

Figure 7-56 illustrates the flow of the business process as follows:

1. The indexer creates a new contract case. The newly created case is saved in the Indexing Inbox. After that, the Indexer sends this case to a reviewer for review.
2. The reviewer can see the assigned cases after logging into the system. The reviewer reviews each case and decides whether to approve or reject it.
3. In addition, the reviewer can hold the case for review later. The case is placed in a personal inbox.
4. If the case passes review, an approver receives the case after logging in. The approver reviews the case and approves or rejects it. If the approver approves the case, the whole process is completed. Otherwise the case is terminated.

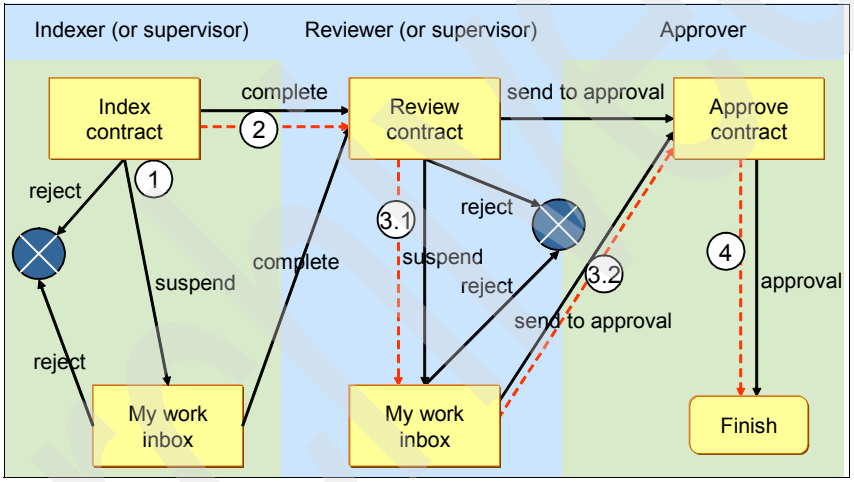


Figure 7-56 Business process

Walking through the Contract Management Workflow

In this section, we take you through the following steps of the workflow:

- Step 1: The indexer creates a new case
- Step 2: The indexer sends the case to review
- Step 3a: The reviewer holds the incoming case
- Step 3b: The reviewer finishes the review and sends case to the approver
- Step 4: The approver approves the case

Step 1: The indexer creates a new case

Complete the following steps:

1. To launch the Contract Management Workflow, launch the **FileNet Workplace** and log in as user ID `joe` with a password of `password`.
2. As shown in Figure 7-57, select **Browse** → **EVTF** and click the **Workflows** folder.

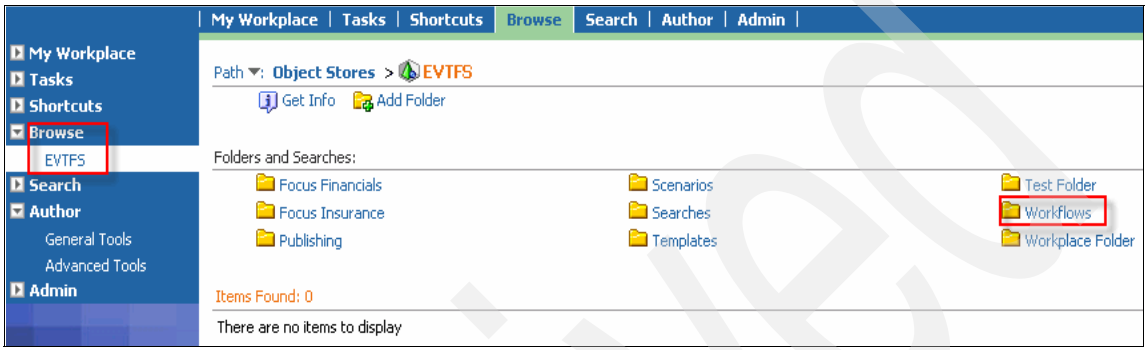


Figure 7-57 Browsing the Workflows folder

Archived

3. Right-click **Contract Management Workflow** and select **Launch** (Figure 7-58).

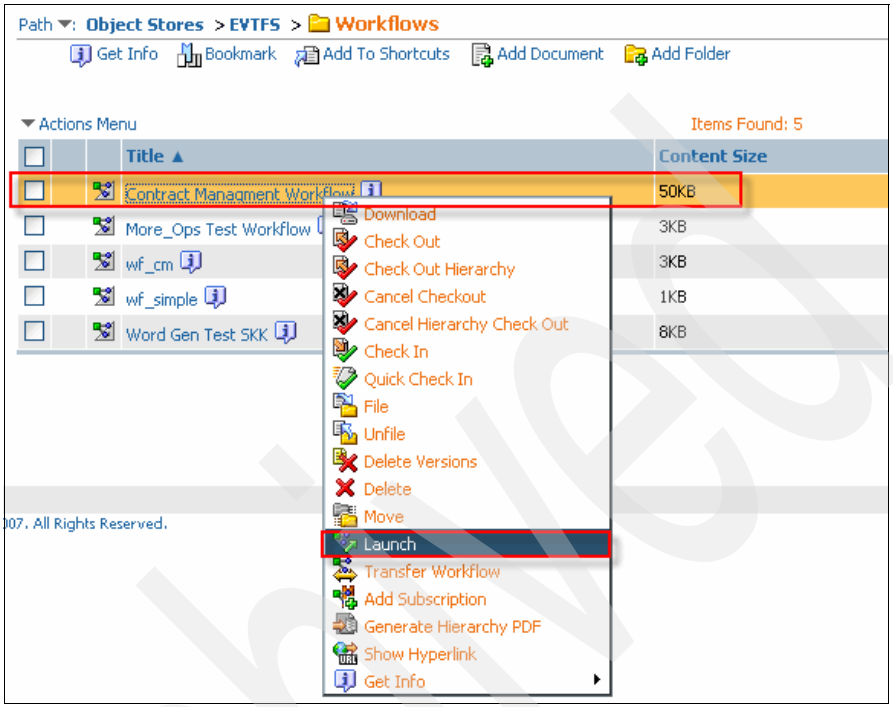


Figure 7-58 Selecting the Launch option

4. In the Launch Workflow panel (Figure 7-59), click **Launch**.

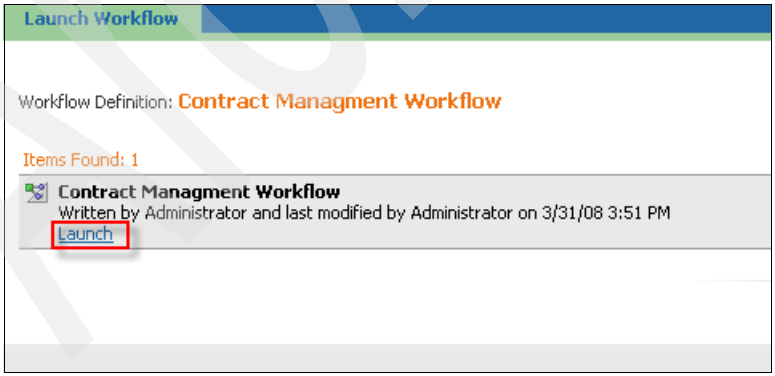


Figure 7-59 Launch Workflow panel

Archived

5. In the Launch Step panel (Figure 7-60), click **Data Fields**.

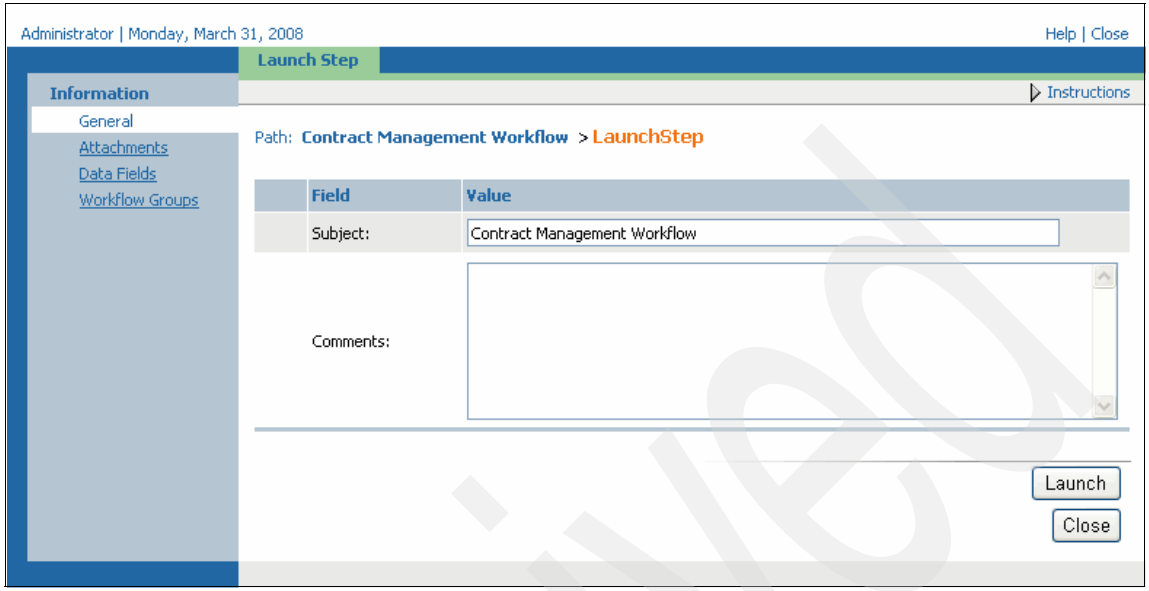


Figure 7-60 Launch Step panel

Archived

6. In the Data Fields panel (Figure 7-61), set AccountNumber to 0001, CompanyName to XYZ, ContractAmount to 1000, and Priority to 1. Click **Launch**.

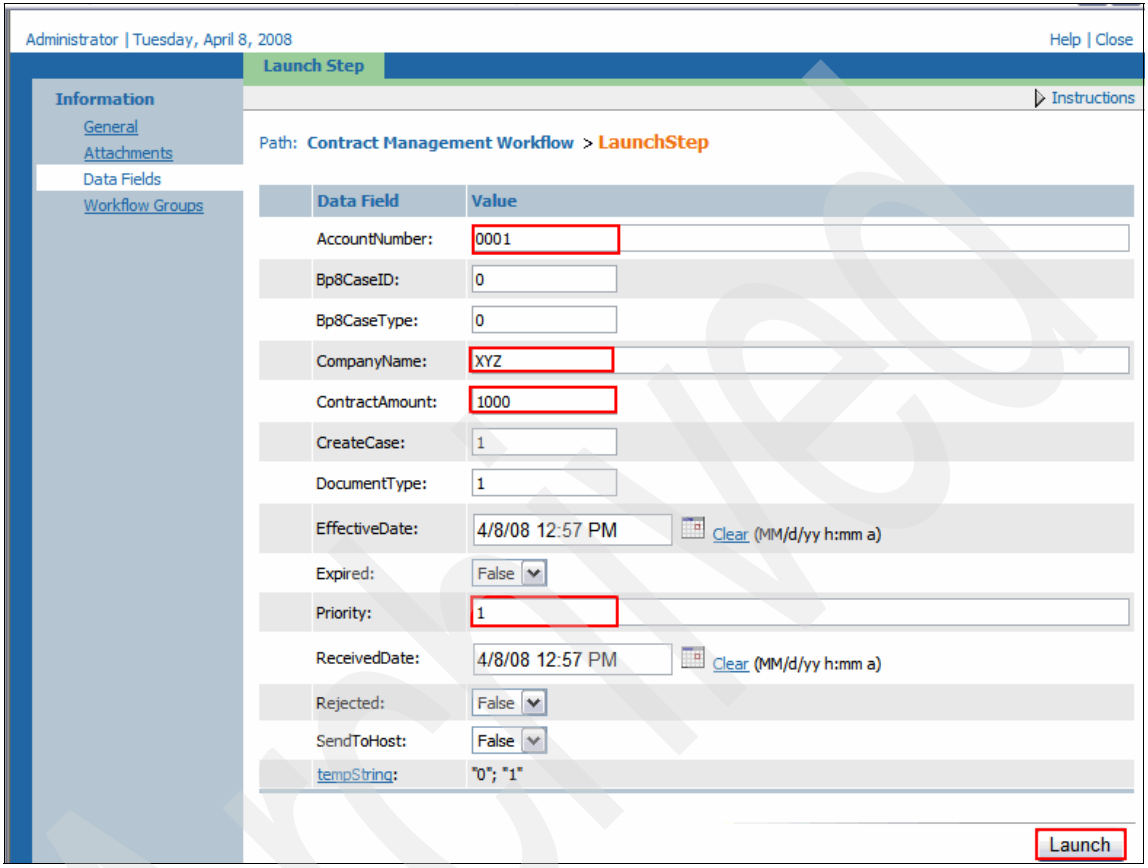


Figure 7-61 Data Fields panel

A new case is created and saved into the Indexing In basket.

Archived

Step 2: The indexer sends the case to review

Complete the following steps:

1. As shown in Figure 7-62, select **Tasks** → **Public Inboxes** and click **Indexing**.

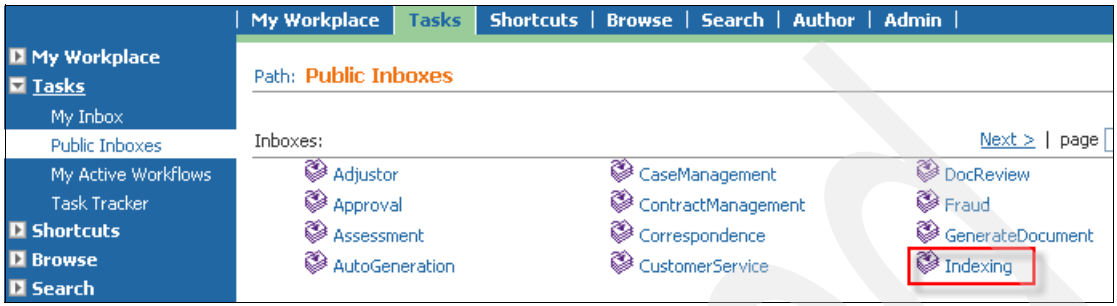


Figure 7-62 Public Inboxes panel

2. In the Public Inboxes panel, click the **Contract Management Workflow** link to launch the workflow (Figure 7-63).

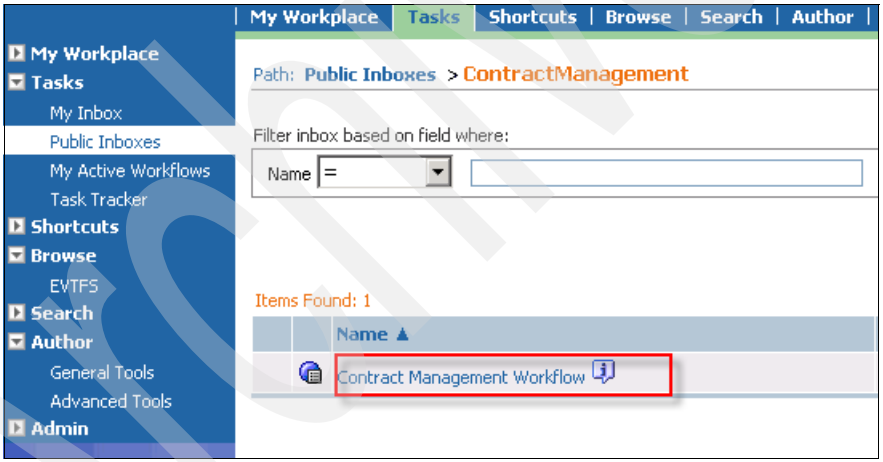


Figure 7-63 Contract Management panel

Archived

3. In the Indexing Queue panel (Figure 7-64), for Responses, select **IndexComplete** and click **Complete**.

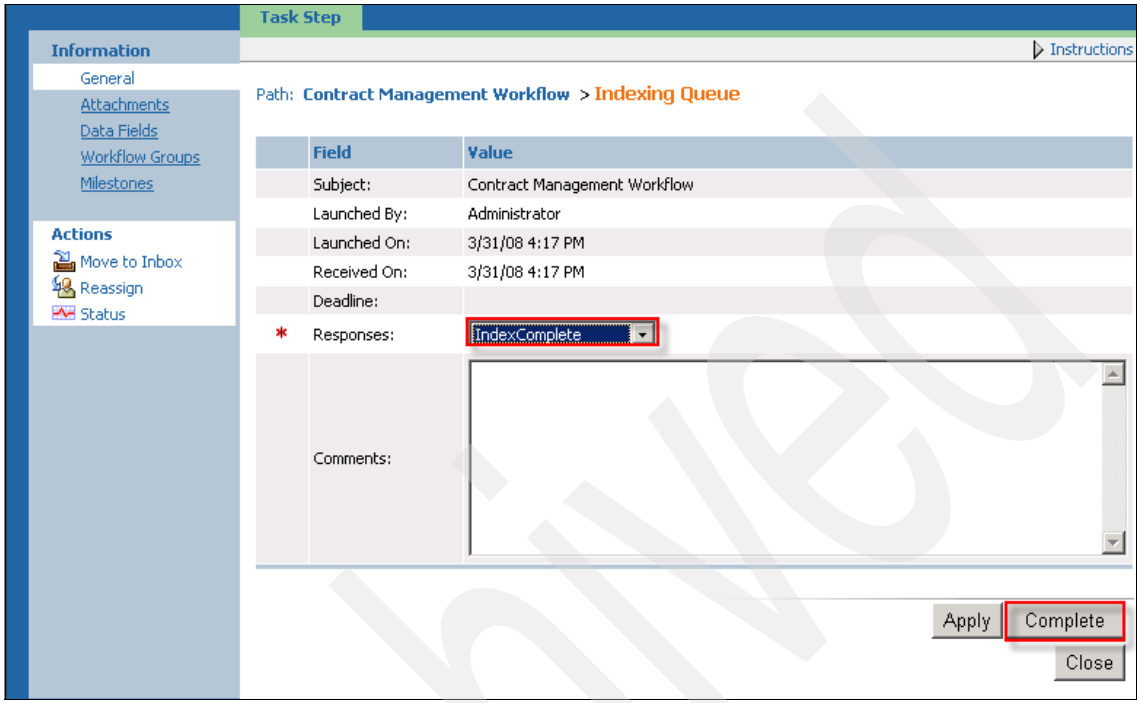


Figure 7-64 Indexing Queue panel

Step 3a: The reviewer holds the incoming case

Complete the following steps:

1. Log in with the user ID `mark` and a password of `password`.
2. Select **Tasks** → **Public Inboxes** and click **Review**, which might be displayed in the next panel.
3. Click the **Contract Management Workflow** link to launch the workflow.

Archived

4. In the Review Queue panel (Figure 7-65), for Responses, select **ReviewPend** and click the **Complete** button.

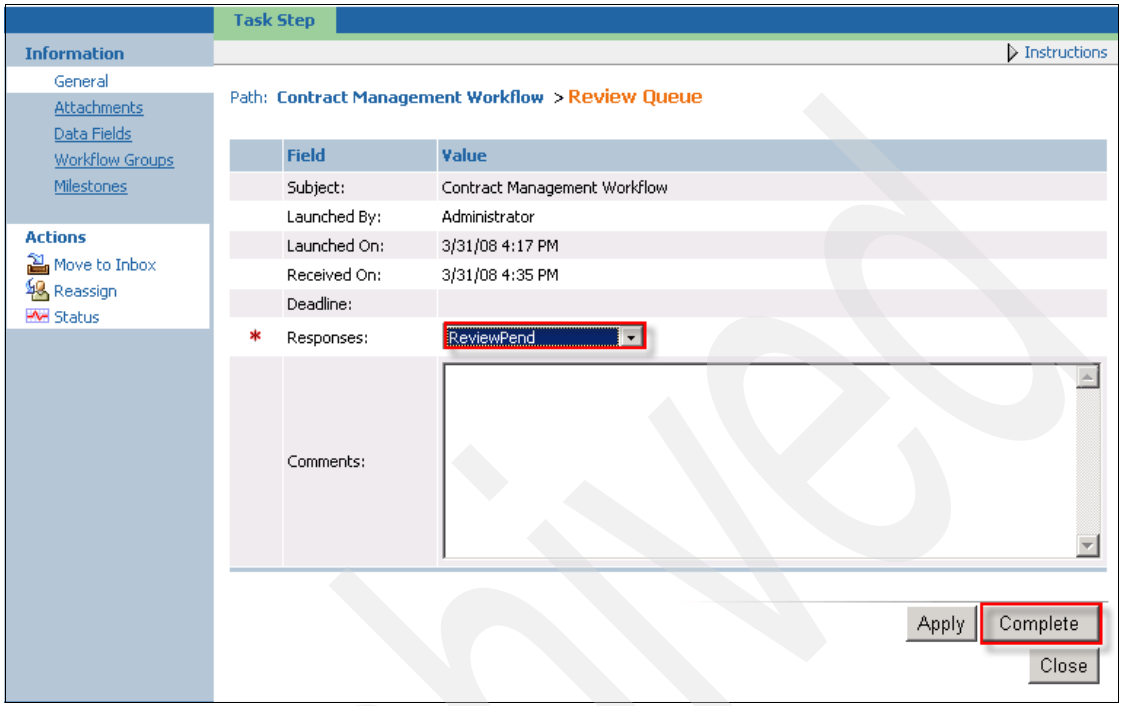


Figure 7-65 Review Queue panel

Archived

Step 3b: The reviewer finishes the review and sends case to the approver

Complete the following steps:

1. As shown in Figure 7-66, select **Tasks** → **My Inbox** and in the right pane, click **Contract Management Workflow**.

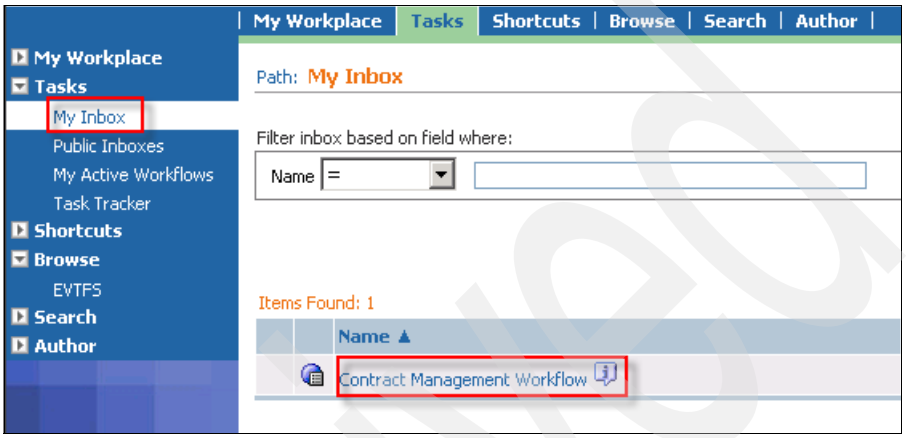


Figure 7-66 My Inbox panel

Archived

2. In the Pend panel (Figure 7-67), for Responses, select **PendSendToApproval** and click the **Complete** button.

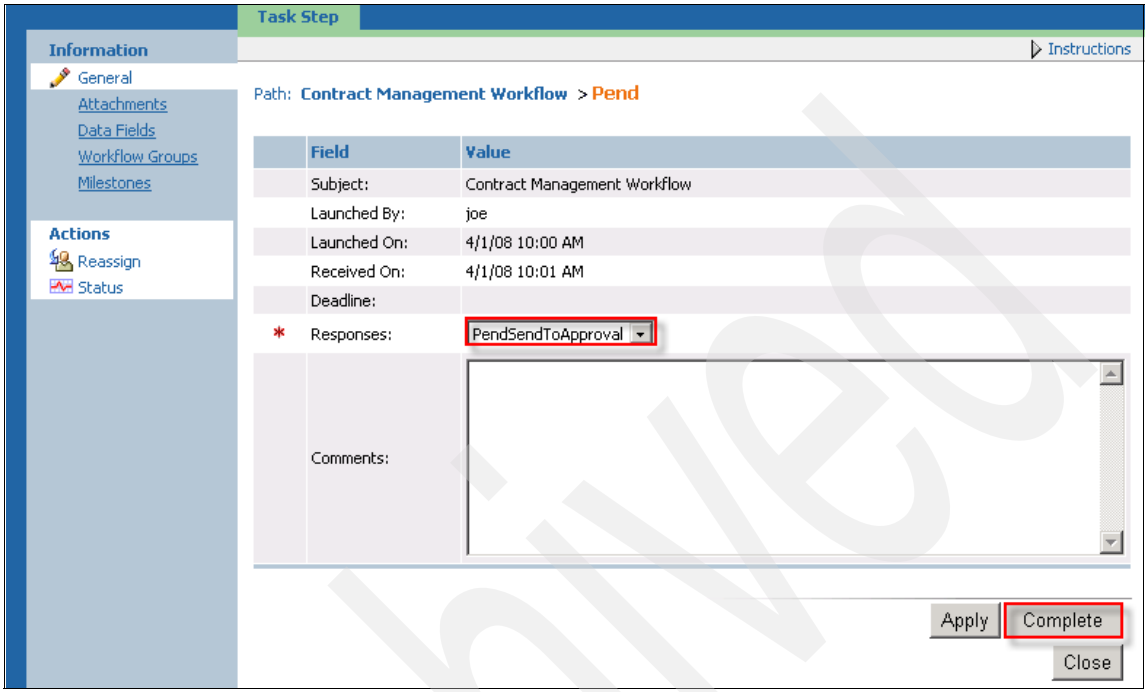


Figure 7-67 Pend panel

Archived

Step 4: The approver approves the case

Complete the following steps:

1. Log in with user ID `ana` and a password of `password`.
2. As shown in Figure 7-68, in the left pane, select **Tasks** → **My Inbox**, and in the right pane, click **Contract Management Workflow**.

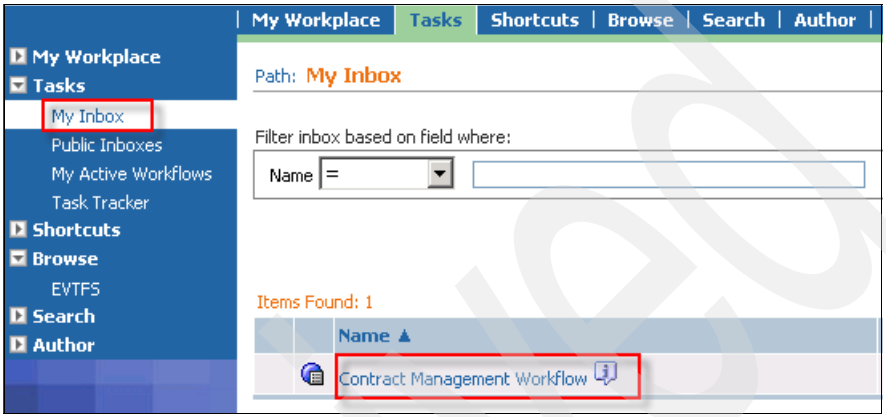


Figure 7-68 My Inbox panel

Archived

3. In the Approve Queue panel (Figure 7-69), for Responses, select **ApproveApprove** and click the **Complete** button.

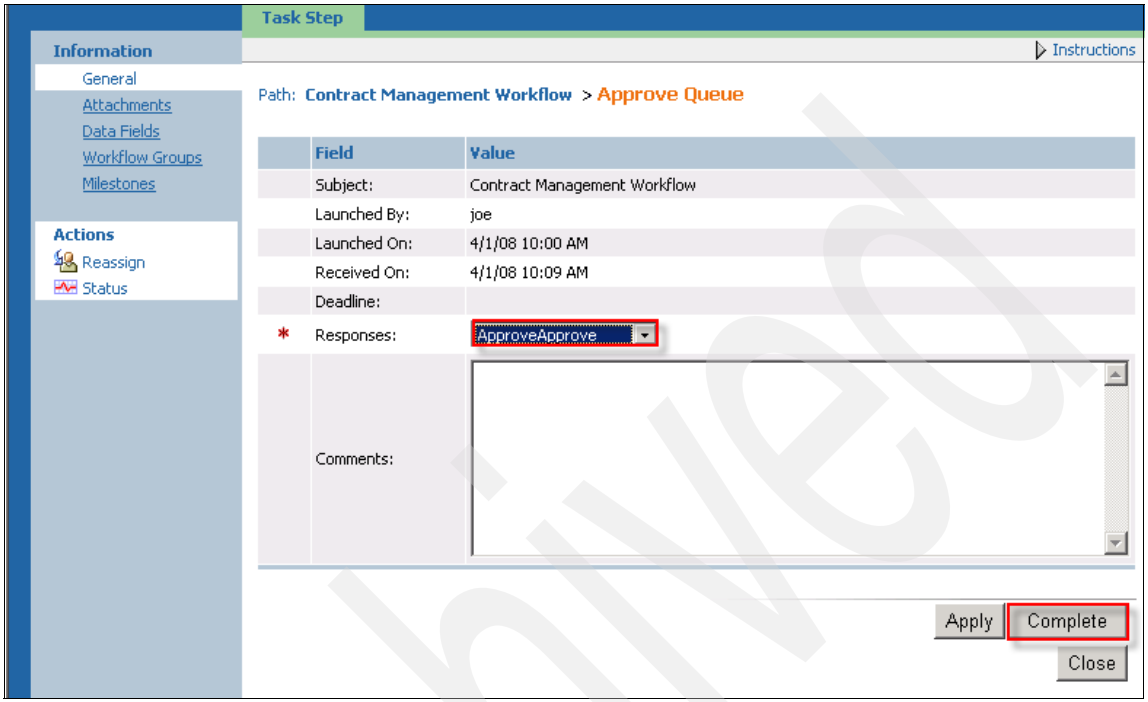


Figure 7-69 Approve Queue

Now, the Contract Management Workflow for AccountNumber 0001 is completed.

Archived

7.9 Creating the FileNet P8 BPM Monitor Model for WebSphere Business Monitor

In this section, we define Contract Management KPIs and measures. We also explain how to create the monitor model.

7.9.1 ClipsAndTacks KPIs and measures

The ClipsAndTacks management team wants the ability to measure the results of the Contract Management Workflow when it is implemented to ensure that it is helping to meet the company's business objectives. To measure this workflow, management has identified business measures for the process:

- KPIs
  - Average total process duration with a target of four days or less and a range between zero and seven days
  - Average total activity duration with a target of one day or less and a range between zero and three days
- Measures
  - Contract amount total and contract amount average
- Dimensions
  - Company
  - Task
  - User
- Alerts
  - Create a situation event if the contract fulfillment time is greater than four days

7.9.2 Importing the monitor model

The FileNet P8 BPM Common Base Event Adapter includes the FileNet P8 BPM Monitor Model, which can be imported into WebSphere Integration Developer with the WebSphere Business Monitor Toolkit. We use this monitor model as a starting point of the Contract Management Workflow monitor model. To import the FileNet P8 BPM Monitor Model:

1. Open WebSphere Integration Developer. Make sure the Business Monitor perspective is active.
2. In the WebSphere Business Monitor Toolkit, right-click in the Project Explorer view and then select **Import**.

Archived

3. In the window that opens, select **Other** → **Project Interchange** and click **Next**.
4. In the Import Projects Interchange Contents window (Figure 7-70):
   a. For From zip file, click **Browse** to find the location of the file, for example, C:\P8PE402WIN\WASMonitor. Select the **BPMMonitoring.zip** file.

      Note: BPMMonitoring.zip is supplied in FileNet P8 FixPack 4.0.2.

   b. Select the **BPMMonitoring** check box.
   c. Click **Finish**.

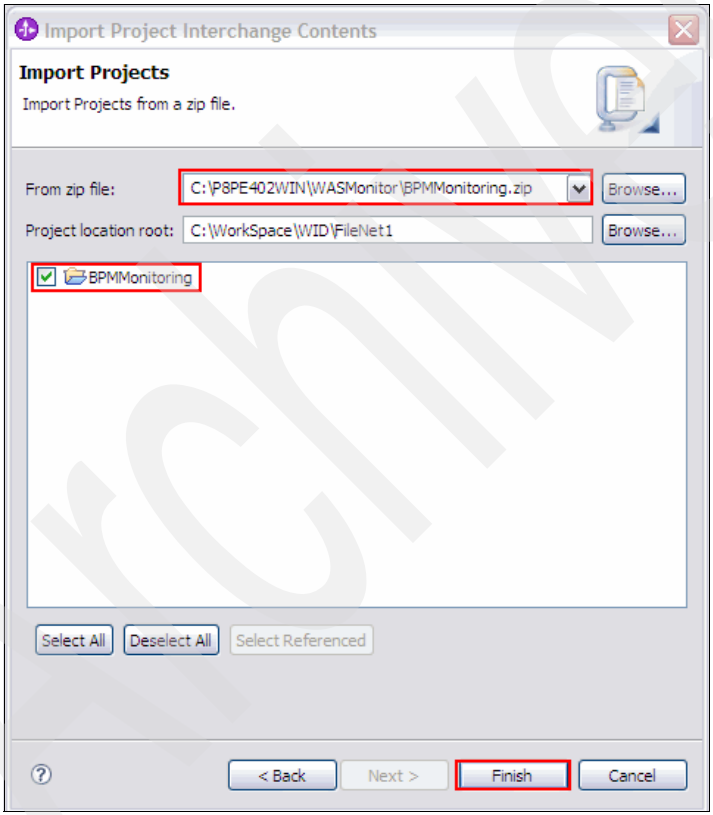


Figure 7-70 Import Project Interchange Contents window

5. In the Project Explorer view, expand **BPMMonitoring** → **Monitor Models** and double-click **BPM Monitoring.mm**.
6. You see an error message. When prompted to create a backup, click **Yes**.

Archived

7. In the Monitor Model Migration window, click **Yes** to migrate the model.

> **Note:** The FileNet P8 BPM Monitor Model (BPMMonitoring.zip) is based on the monitor model of WebSphere Business Monitor V6.0.2. Therefore, we must migrate the monitor model after importing the monitor model project.

The Monitor Model editor opens (Figure 7-71) and shows the following tabs:

- The Monitor Details Model tab shows all monitoring contexts and events.
- The KPI Model tab shows the KPIs (none at this point).
- The Dimensional Model tab shows the cube dimensions and measures.
- The Visual Model tab shows the diagram (if we copy the Scalable Vector Graphics (SVG) files).
- The Event Model tab shows the events.
- The BPMMonitoring.mm tab shows the underlying XML file.

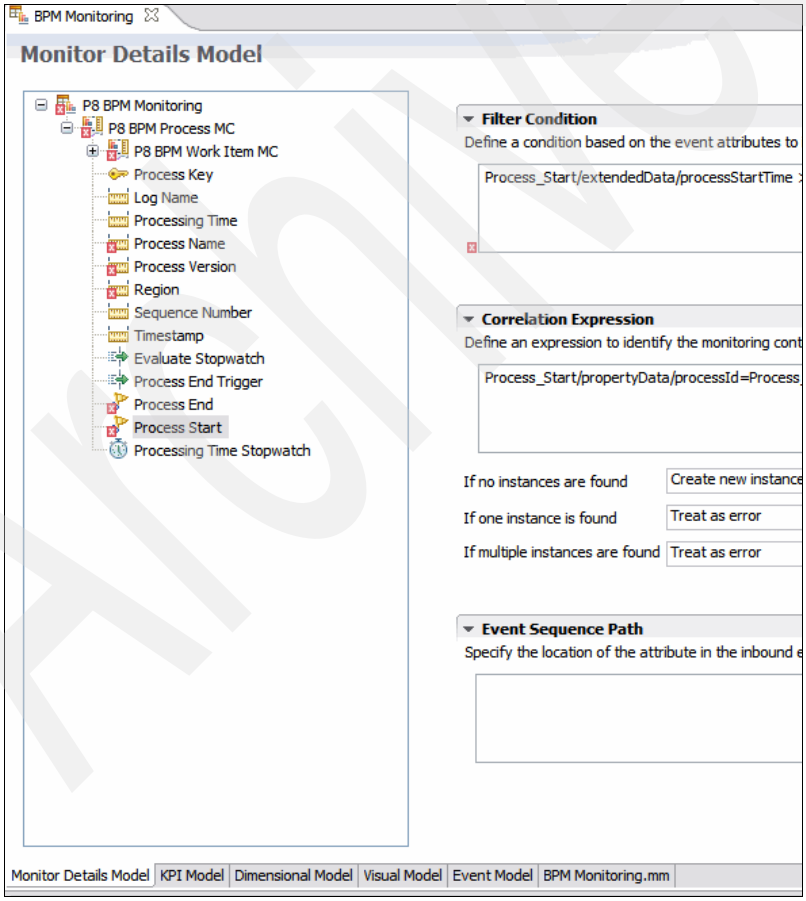


Figure 7-71 Monitor Details Model panel

8. Select **Project** → **Clean**, and then select **Clean all projects**. Click **OK**.

> **Note:** You see error messages in the monitor model. We solve these error messages later.

9. In the Project Explorer, expand **BPMMonitoring** → **Event Definitions**. You see the predefined process and activity events (Figure 7-72). The details of these events are provided in 7.5, “FileNet P8 BPM Common Base Event events” on page 381.

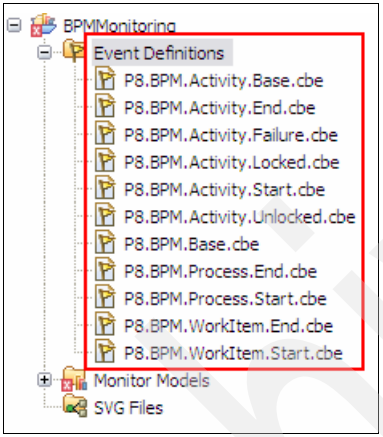


Figure 7-72 Event Definitions

Archived

The monitor model has two types of context (Figure 7-73):

– *Process level context* defines how to monitor process level events.
– *Work item level context* defines how to monitor work item level events.

Note: The details of these monitor contexts are provided in 7.6, “FileNet P8 BPM Monitor Model for WebSphere Business Monitor” on page 384.

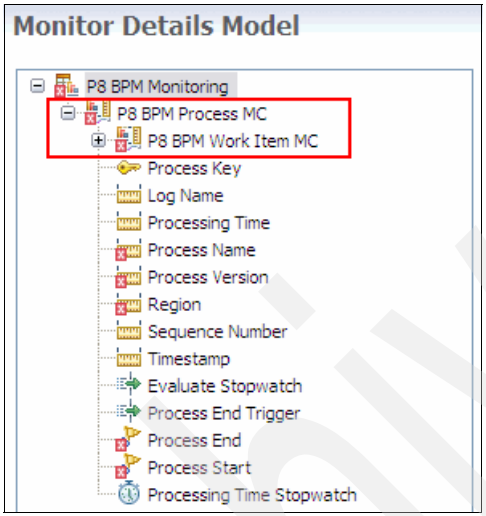


Figure 7-73 Monitor contexts in the Monitor Details Model panel

Archived

10. To configure the event filter for the inbound event so that the event's Process Start Time is greater than Monitor Model's Deploy DateTime, select **P8 BPM Monitoring** → **P8 BPM Process MC** → **Process Start** (Figure 7-74).

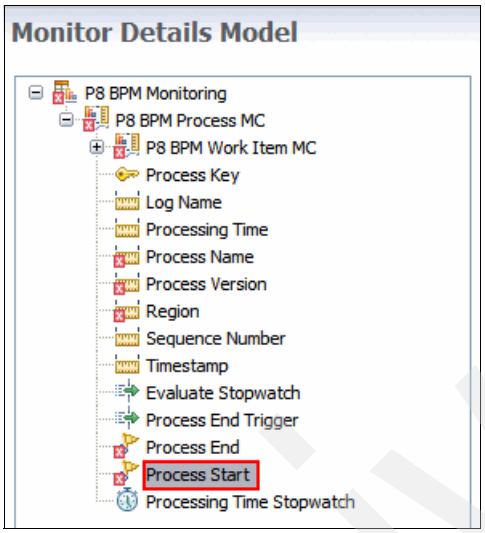


Figure 7-74 Process Start inbound event

11. Set the Filter Condition to the following value as shown in Figure 7-75:

```
Process_Start/extendedData/processStartTime >
dateTime('2007-07-31T14:47:27-07:00') and
Process_Start/extendedData/processName = 'Contract Management
Workflow'
```

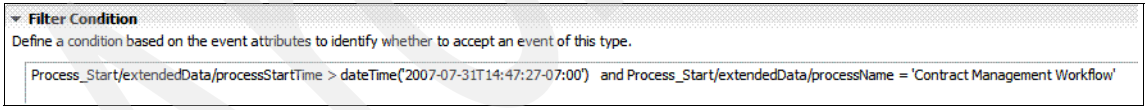


Figure 7-75 Filter Condition

This setting specifies that only processes of Contract Management Workflow that start after the specified time will be considered by the monitor model.

Archived

12. Repeat steps 10 and 11 for the following inbound events. Each time, you must replace *Process_Start* with the relevant inbound event name, for example *Process_End*:
    - P8 BPM Process MC → Process End
    - P8 BPM Process MC → P8 BPM Work Item MC → Activity End
    - P8 BPM Process MC → P8 BPM Work Item MC → Activity Failure
    - P8 BPM Process MC → P8 BPM Work Item MC → Activity Locked
    - P8 BPM Process MC → P8 BPM Work Item MC → Activity Start
    - P8 BPM Process MC → P8 BPM Work Item MC → Activity Unlocked
    - P8 BPM Process MC → P8 BPM Work Item MC → WorkItem End
    - P8 BPM Process MC → P8 BPM Work Item MC → WorkItem Start
13. Press Ctrl+S to save your work. There is no future inbound event error messages, although others will persist.
14. To set the default value of the metric used in the Dimensional Model, select **P8 BPM Monitoring → P8 BPM Process MC → Process Name**.
15. Change the Default Value to an empty string (two single quotation marks) as shown in Figure 7-76.

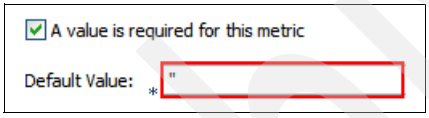


Figure 7-76 Default Value

16. Change the Default Value of the following metrics as described in Table 7-12.

Table 7-12 The Default Value of the metrics

Metric	Default Value
P8 BPM Process MC →Process Version	'1.0'
P8 BPM Process MC →Process Region	1
P8 BPM Process MC → P8 BPM Work Item MC →Process Name	''
P8 BPM Process MC →P8 BPM Work Item MC →Region	1
P8 BPM Process MC →P8 BPM Work Item MC →User	''

17. Press Ctrl+S to save your work. There is no error messages in the P8 BPM Monitoring monitor model.

Archived

7.9.3 Updating the event definition

To add the data fields CompanyName, ContractAmount, and AccountNumber to the P8.BPM.Base event:

1. In Project Explorer, expand **BPMMonitoring** → **Event Definitions** → **P8.BPM.Base.cbe** (Figure 7-77).

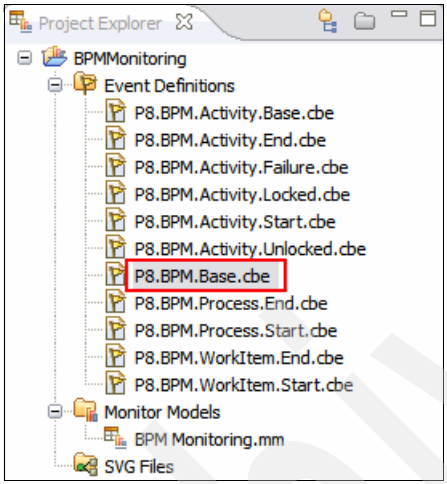


Figure 7-77 Project Explorer

Archived

2. In the editor (Figure 7-78), add three elements as children of businessData:
   a. Click **businessData** and then click the **Add Extended Data As Nested Child** icon.
   b. Name the new child element `CompanyName` and set its type to **string**.
   c. Repeat these steps to create the following child elements:
      - Child element named `ContractAmount` with a type of **float**
      - Child element named `AccountNumber` with a type of **string**

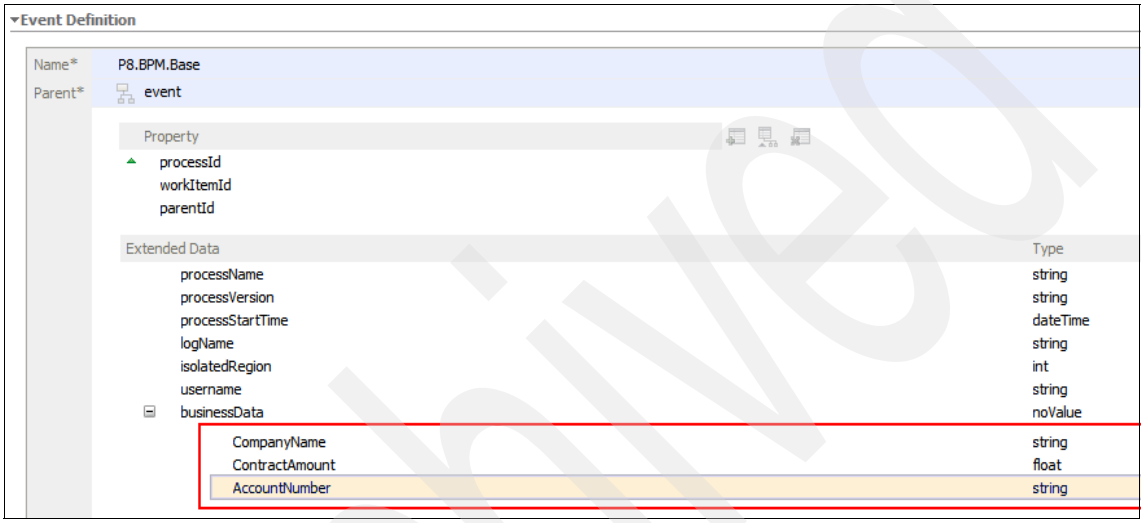


Figure 7-78 Event Definition

3. Press Ctrl+S to save your work

7.9.4 Creating metrics

In this section, we explain how to create the CompanyName, ContractAmount, and AccountNumber metrics at the process level.

Archived

Creating the CompanyName metric

To create the CompanyName metric:

1. In the Monitor Details Model panel (Figure 7-79), in the model tree, right-click **P8 BPM Process MC** and select **New** → **Metric**.

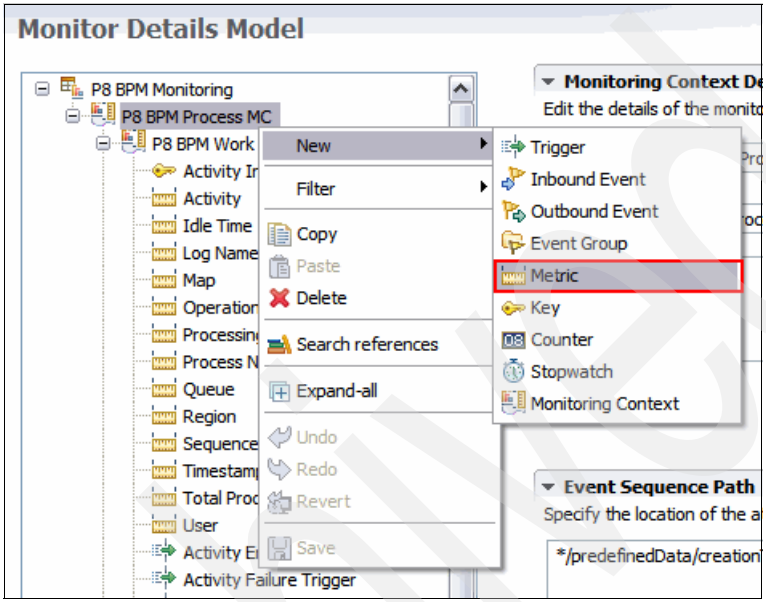


Figure 7-79 Monitor Details Model

2. In the Create New Metric window (Figure 7-80), for Name and ID, type `CompanyName`, and for Type, select **String**. Then click **OK**.

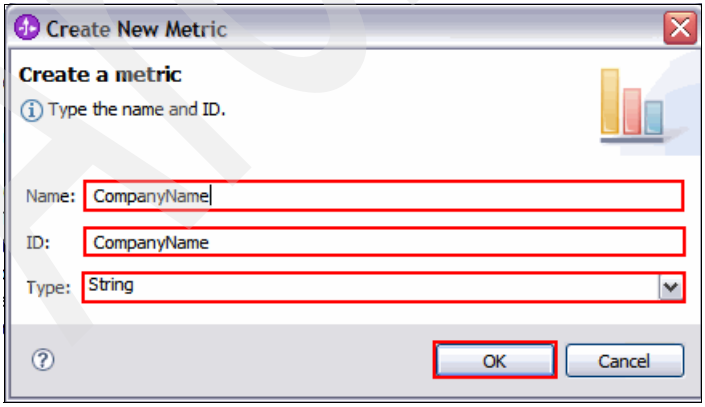


Figure 7-80 Create New Metric window

3. In the next panel (Figure 7-81):
   a. Select **A value is requires for this metric** since a dimension uses this metric and that requires a value for dimensional analysis.
   b. For Default Value, type an empty string (two single quotation marks).
   c. Under Metric Value Expressions, click the **Add** button. In the Expression cell, use content assist to navigate to the `Process_Start/extendedData/businessData/CompanyName` inbound event field for CompanyName.

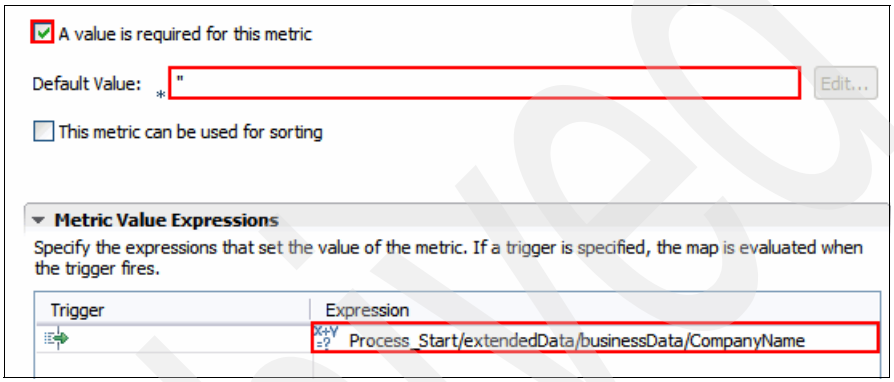


Figure 7-81 Metric details

Creating the ContractAmount metric

To create the ContractAmount metric:

1. In the Monitor Details Model panel, right-click **P8 BPM Process MC** in the model tree, and select **New** → **Metric**.
2. In the Create New Metric window, for Name and ID, type `ContractAmount`, and for Type, select **Decimal**. Then click **OK**.
3. In the Metric Value Expressions, click the **Add** button. In the Expression cell, use content assist to navigate to the `Process_Start/extendedData/businessData/ContractAmount` inbound event field for ContractAmount.

Creating the AccountNumber metric

To create the AccountNumber metric:

1. In the Monitor Details Model panel, right-click **P8 BPM Process MC** in the model tree, and select **New** → **Metric**.
2. In the Create New Metric window, for Name and ID, type `AccountNumber`, and for Type, select **String**. Then click **OK**.

Archived

3. In the Metric Value Expressions, click **Add** the button. In the Expression cell, use content assist to navigate to the `Process_Start/extendedData/businessData/AccountNumber` inbound event field for AccountNumber.
4. Press Ctrl+S to save all the metrics that you just created.

7.9.5 Creating measures

Next we explain how to define the Contract Amount Average and Contract Amount Total measures.

Creating the Contract Amount Average measure

To create the Contract Amount Average measure:

1. In the Dimensional Model panel (Figure 7-82), right-click **P8 BPM Process MC Cube** in the model tree, and select **New** → **Measure**.

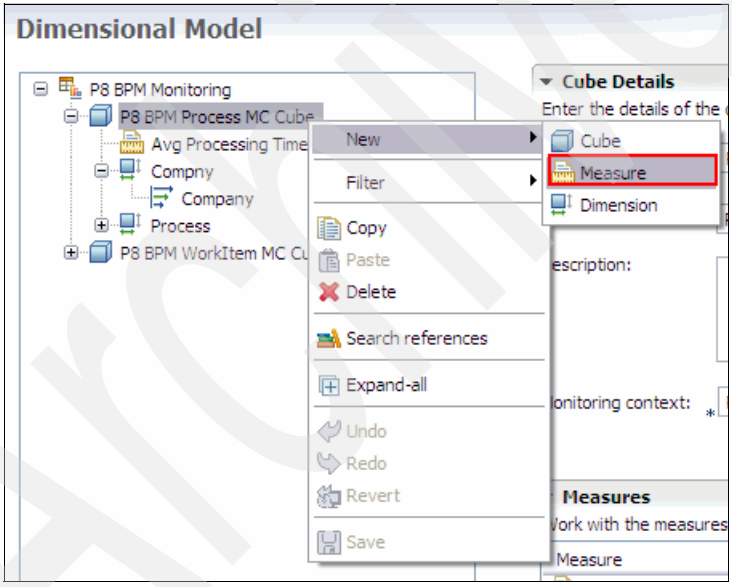


Figure 7-82 Dimensional Model page

Archived

2. In the Create New Measure window (Figure 7-83):
   a. For Name, type `ContractAmountAverage`.
   b. For Source metric, click **Browse**. Expand **P8 BPM Process MC** and select **ContractAmount**. Click **OK**.
   c. For Aggregation function, select **Average**.
   d. Click **OK**.

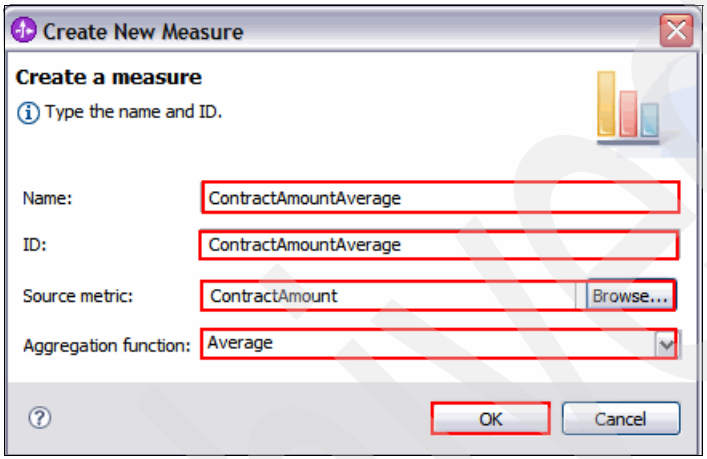


Figure 7-83 Create New Measure window

Creating the Contract Amount Total measure

To create the Contract Amount Total measure:

1. In the Dimensional Model panel, right-click **P8 BPM Process MC Cube** in the model tree, and select **New** → **Measure**.
2. In the Create New Measure window:
   a. For Name, type `ContractAmountTotal`.
   b. For Source metric, click **Browse**. Expand **P8 BPM Process MC** and select **ContractAmount**. Click **OK**.
   c. For Aggregation function, select **Sum**.
   d. Click **OK**.

Archived

Creating the Avg Processing Time (Process) measure

To create the Avg Processing Time (Process) measure:

1. In the Dimensional Model panel, right-click **P8 BPM Process MC Cube** in the model tree and select **New** → **Measure**.
2. In the Create New Measure window:
   a. For Name, type `Avg Processing Time`.
   b. For Source metric, click **Browse**. Expand **P8 BPM Process MC** and select **Processing Time Stopwatch**. Click **OK**.
   c. For Aggregation function, select **Average**.
   d. Click **OK**.
3. Press Ctrl+S to save the measures that you just created.

7.9.6 Creating dimensions

After importing the FileNet BPM Monitor Model, two cubes are defined in the Data mart model:

- BPM Process MC Cube (Cube for Process)
- BPM Work Item MC Cube (Cube for Work Item)

Note: The details of these cubes are provided in 7.6, “FileNet P8 BPM Monitor Model for WebSphere Business Monitor” on page 384.

In the sections that follow, we explain how to define the Company, Task, and User dimensions.

Archived

Creating the Company dimension

A Company dimension is useful for drilling down in the Dimensional view of the dashboard to aggregate instances by the company. To create it:

1. In the Dimensional Model panel (Figure 7-84), right-click **P8 BPM Process MC Cube** in the model tree, and select **New** → **Dimension**.

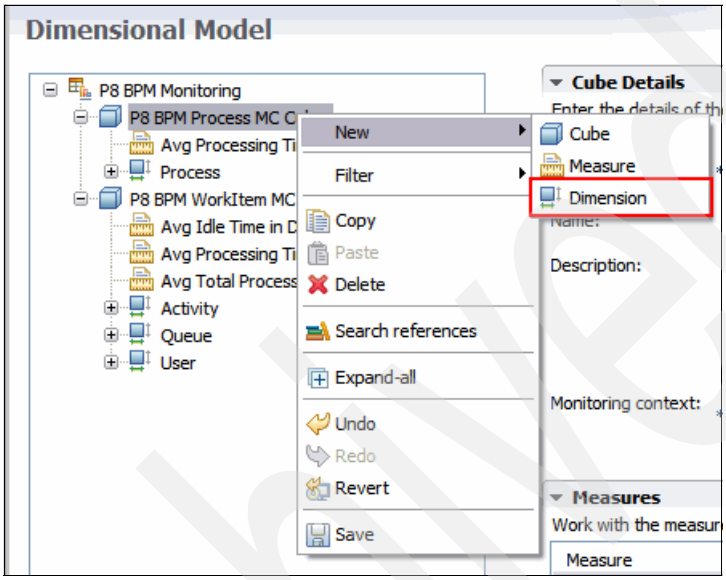


Figure 7-84 Dimensional Model

2. In the Create New Dimension window (Figure 7-85), for Name, type `Company`, and click **OK**.

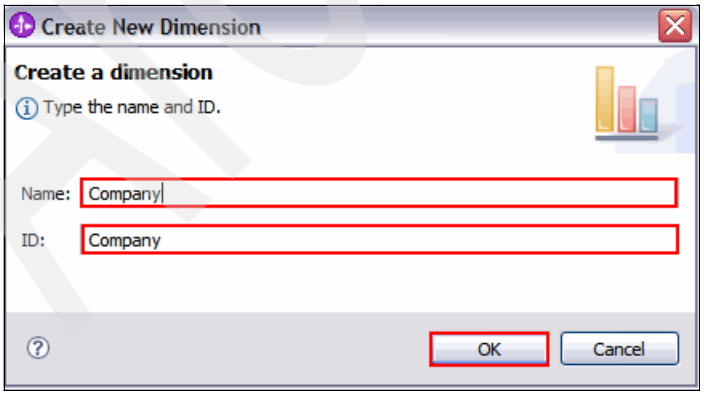


Figure 7-85 Create New Dimension window

Archived

3. In the Dimensional Model navigator, right-click **Company** and select **New** → **Dimension Level**.
4. In the Create New Level window (Figure 7-86):
   a. For Name, type `Company`.
   b. For Source metric, click **Browse**.
   c. Expand **P8 BPM Process MC** and select **CompanyName**. Click **OK**.
   d. Click **OK**.

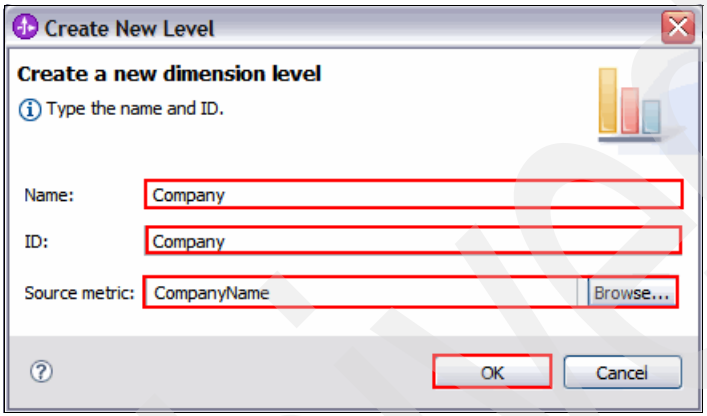


Figure 7-86 Create New Level window

Creating the Task dimension

Creating a Task dimension is useful for drilling down in the Dimensional view of the dashboard to aggregate instances by the map and activity. To create this dimension:

1. In the Dimensional Model panel, right-click the **P8 BPM WorkItem MC Cube** in the model tree, and select **New** → **Dimension**.
2. In the Create New Metric window, for Name, type `Task` and click **OK**.
3. In the Dimensional Model navigator, right-click **Task** and select **New** → **Dimension Level**.
4. In the Create New Level window:
   a. For Name, type `Map`.
   b. For Source metric, click **Browse**.
   c. Expand **P8 BPM WorkItem MC** and select **Map**. Click **OK**.
   d. Click **OK**.
5. Create a second Dimension level. In the Dimensional Model navigator, right-click **Task** and select **New** → **Dimension Level**.

Archived

6. In the Create New Level window:
   a. For Name, type `Activity`.
   b. For Source metric, click **Browse**.
   c. Expand **P8 BPM WorkItem MC** and select **Activity**. Click **OK**.
   d. Click **OK**.
7. Press Ctrl+S to save the dimensions that you just created.

Creating the User dimension

When you import the P8 BPM Monitor Model in to your workspace, you see the User dimension under P8 BPM WorkItem MC Cube by default.

7.9.7 Creating inbound events

To create Total Processing Time KPIs of each activity, we must create the inbound events of the start and stop activity filtered by the activity name. In this section, we define the inbound events of Indexing Queue Activity.

Creating the Indexing Queue Activity Start inbound event

To create the Indexing Queue Activity Start inbound event:

1. In the Monitor Details Model panel, in the model tree, expand **P8 BPM Process MC**, right-click **P8 BPM Work Item MC**, and select **New** → **Inbound Event**.
2. In the Create New Inbound Event window, for Name, type `Indexing Queue Activity Start` and click **OK**.
3. Under Event Type Details (Figure 7-87), for Extension name, click **Browse**. Expand **BPMMonitoring** and select **P8.BPM.Activity.Start.cbe** and select **P8.BPM.Activity.Start**. Click **OK**.

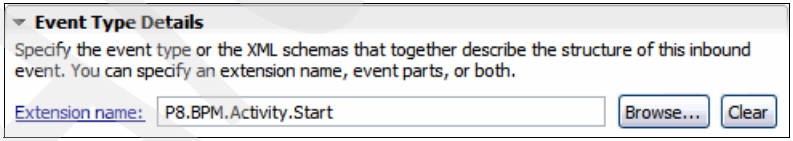


Figure 7-87 Event Type Details

Archive

4. Under Filter Condition (Figure 7-88), type the following condition:

```
Indexing_Queue_Activity_Start/extendedData/processStartTime >
dateTime('2007-07-31T14:47:27-07:00') and
Indexing_Queue_Activity_Start/extendedData/processName = 'Contract
Management Workflow' and
Indexing_Queue_Activity_Start/extendedData/map = 'Indexer Process'
and Indexing_Queue_Activity_Start/extendedData/activity =  'Indexing
Queue'
```

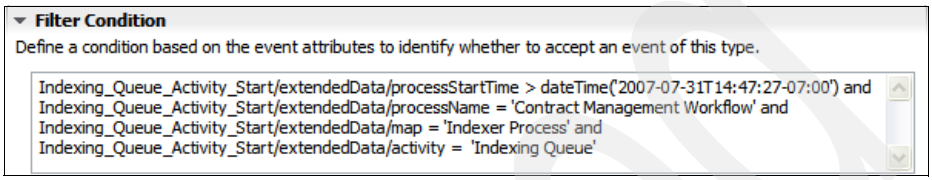


Figure 7-88 Filter Condition

5. Under Correlation Expression (Figure 7-89):
   a. Type the following Correlation Expression:

```
Indexing_Queue_Activity_Start/propertyData/workItemId =
Activity_Instance_Id and
Indexing_Queue_Activity_Start/propertyData/processId =
../Process_Key
```

   b. For If no instances are found, select **Ignore**.
   c. For If one instance is found, select **Deliver to the instance**.
   d. For If multiple instances are found, select **Treat as error**.

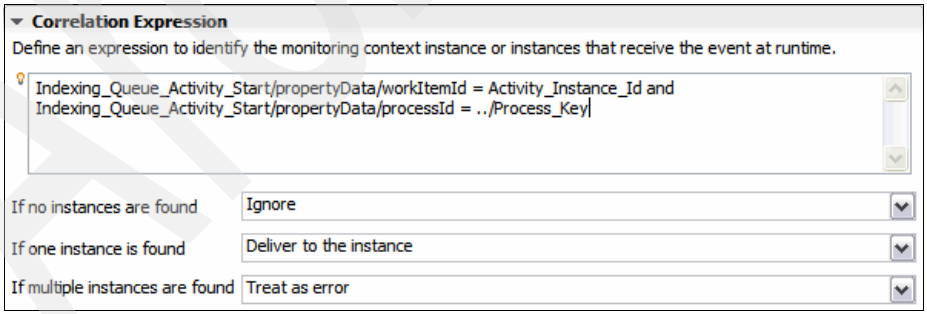


Figure 7-89 Correlation Expression

Archived

Creating an Indexing Queue Activity End inbound event

To create an Indexing Queue Activity End inbound event:

1. Under the Monitor Details Model tab, in the model tree, expand **P8 BPM Process MC**. Right-click **P8 BPM Work Item MC** and select **New** → **Inbound Event**.
2. In the Create New Metric window, for Name, type `Indexing Queue Activity End` and click **OK**.
3. Under Event Type Details, for Extension name, click **Browse**. Expand **BPMMonitoring** → **P8.BPM.Activity.End.cbe**. Select **P8.BPM.Activity.End**.
4. Under Filter Condition, enter the following condition:

```
Indexing_Queue_Activity_End/extendedData/processStartTime >
dateTime('2007-07-31T14:47:27-07:00') and
Indexing_Queue_Activity_End/extendedData/processName = 'Contract
Management Workflow' and
Indexing_Queue_Activity_End/extendedData/map = 'Indexer Process' and
Indexing_Queue_Activity_End/extendedData/activity =  'Indexing
Queue'
```

5. Under Correlation Expression:
   a. Enter the following Correlation Expression:

```
Indexing_Queue_Activity_End/propertyData/workItemId =
Activity_Instance_Id and
Indexing_Queue_Activity_End/propertyData/processId =
../Process_Key
```

   b. For If no instances are found, select **Ignore**.
   c. For If one instance is found, select **Deliver to the instance**.
   d. For If multiple instances are found, select **Treat as error**.
6. Press Ctrl+S to save the events that you just created.

Creating Start and Stop inbound events for other activities

Repeat the instructions in the previous section to create inbound events for the other activities by using the filter conditions in Table 7-13 on page 436. Each time, you must replace *Indexing_Queue_Activity_Start* and *Indexing_Queue_Activity_End* with the relevant inbound event names, for example *Pended_Index_Queue_Activity_Start* and *Pended_Index_Queue_Activity_End* respectively.

Archived

Table 7-13 Filter conditions

Inbound event	/extendedData/map	/extendedData/activity
Pended Index Queue Activity Start/ Pended Index Queue Activity End	'Indexer Process'	'Pend'
Review Queue Activity Start/ Review Queue Activity End	'Review Process'	'Review Queue'
Pended Review Queue Activity Start/ Pended Review Queue Activity End	'Review Process'	'Pend'
Approve Queue Activity Start/ Approve Queue Activity End	'Approve Process'	'Approve Queue'

7.9.8 Creating stopwatches

In this section, we explain how to define the Total Indexing Queue Processing Time stopwatch under P8 BPM Work Item MC. The stopwatch is controlled by Indexing Queue Processing Active Start and Activity Stop inbound events.

Creating Total Indexing Queue Processing Time stopwatch

To create the Total Indexing Queue Processing Time stopwatch:

1. In the Monitor Details Model panel, in the model tree, expand **P8 BPM Process MC**, right-click **P8 BPM Work Item MC**, and select **New** → **Stopwatch**.
2. In the Create New Metric window, for Name, type `Total Indexing Queue Processing Time Stopwatch` and click **OK**.
3. Under Stopwatch Controls, click the **Add** button.

Archived

4. In the Select Trigger or Inbound Event window (Figure 7-90), select the **Indexing Queue Activity Start** inbound event and click **OK**.

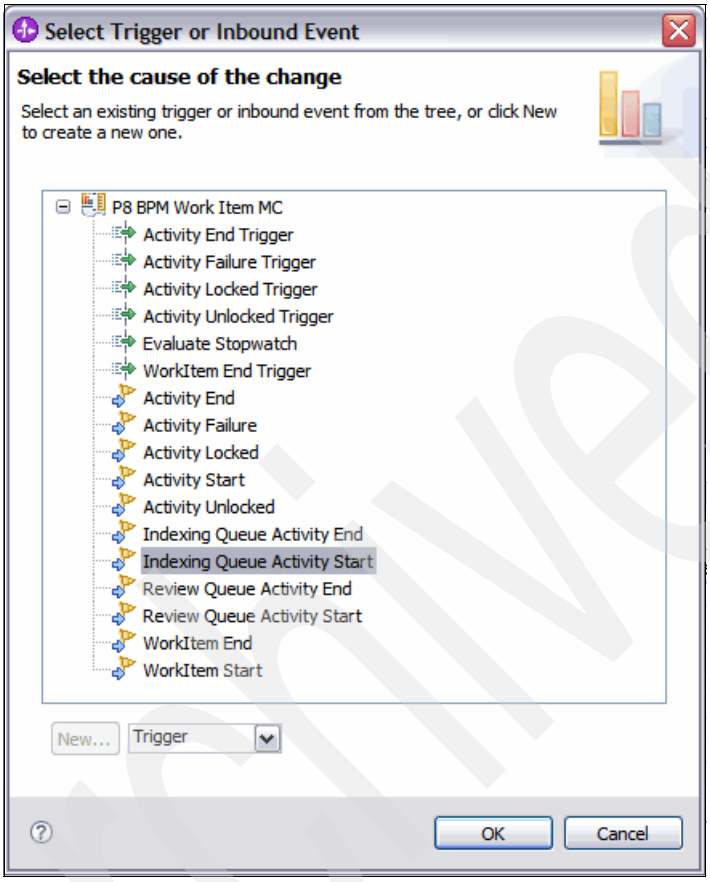


Figure 7-90 Select Trigger or Inbound Event window

5. Under Stopwatch Controls (Figure 7-91), for the Result Action of Activity Start, select **Start**.

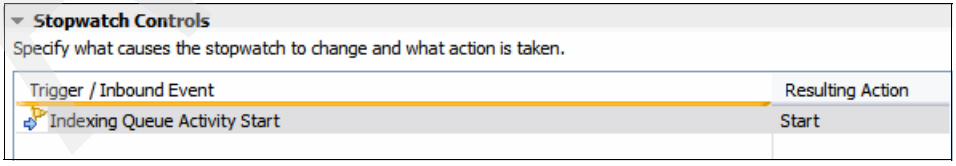


Figure 7-91 Stopwatch Controls - Result Action of Activity Start

6. Under Stopwatch Controls, click the **Add** button.

Archived

7. In the Select Trigger or Inbound Event window, select the **Indexing Queue Activity End** inbound event and click **OK**.
8. Under Stopwatch Controls (Figure 7-92), for the Result Action of Indexing Queue Activity End, select **Stop**.

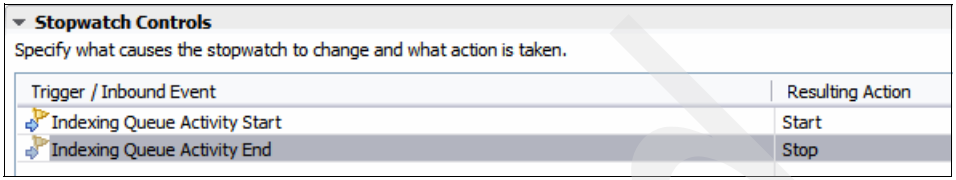


Figure 7-92 Stopwatch Controls - Result Action of Indexing Queue Activity End

9. Press Ctrl+S to save your work.

Creating a stopwatch for other activities

Repeat the instructions in the previous section to create a stopwatch for the other activities by using the Trigger/Inbound Event listed in Table 7-14.

Table 7-14 Trigger/Inbound Event

Stopwatch	Trigger/Inbound Event (Start)	Trigger/Inbound Event (Stop)
Total Pended Index Queue Processing Time Stopwatch	Pended Index Queue Activity Start	Pended Index Queue Activity End
Total Review Queue Processing Time Stopwatch	Review Queue Activity Start	Review Queue Activity End
Total Pended Review Queue Processing Time Stopwatch	Pended Review Queue Activity Start	Pended Review Queue Activity End
Total Approve Queue Processing Time Stopwatch	Approve Queue Activity Start	Approve Queue Activity End

Archived

7.9.9 Creating KPIs

In this section, we explain how to create KPIs for Total Processing Time and Activity Processing time of the Contract Management Workflow.

Creating the Average Total Processing Time KPI

To create the Average Total Processing Time KPI:

1. In the KPI Model panel, in the model tree, right-click **P8 BPM Monitoring** and select **New** → **KPI Context**.
2. In the Create New KPI Context window, for Name, type `Average Total Processing Time` and click **OK**.
3. In the KPI Context navigator, right-click **Average Total Processing Time** and select **New** → **KPI**.
4. In the Create New KPI window, for Name, type `Average Total Processing Time` and click **OK**.
5. In the KPI Details section (Figure 7-93), for Type, select **Duration**.

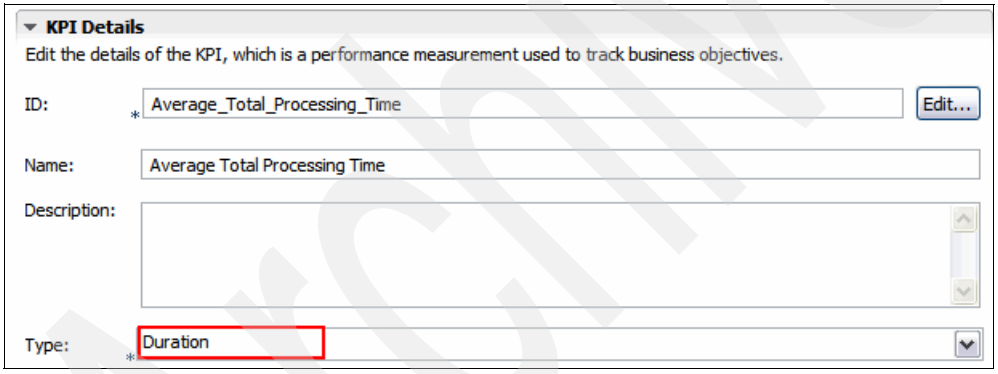


Figure 7-93 KPI Details

6. In the KPI Target and Ranges section (Figure 7-94), for Target, type `Details`.

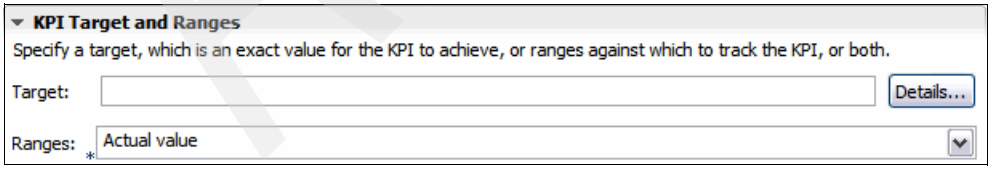


Figure 7-94 KPI Target and Ranges

Archived

7. In the Target Details window (Figure 7-95), for Days Value, type 4 and click **Finish**.

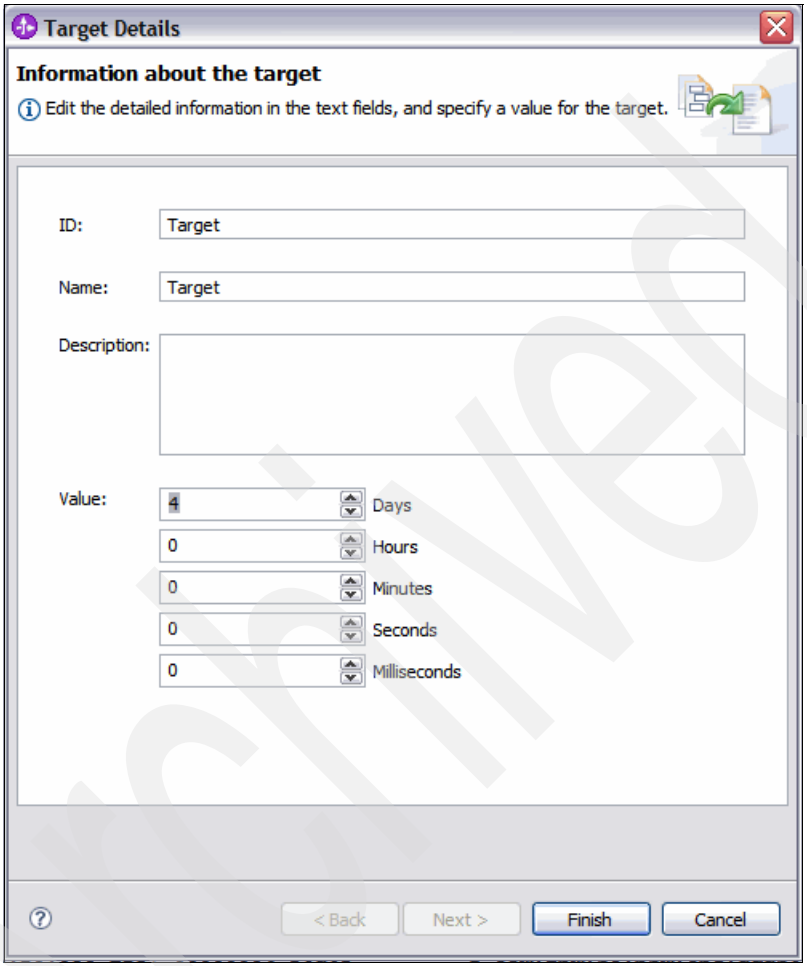


Figure 7-95 Target Details

Archived

8. Under KPI Target and Ranges (Figure 7-96), enter the ranges:
   a. Leave the Range type as **Actual value**. Click **Add** to add a range.

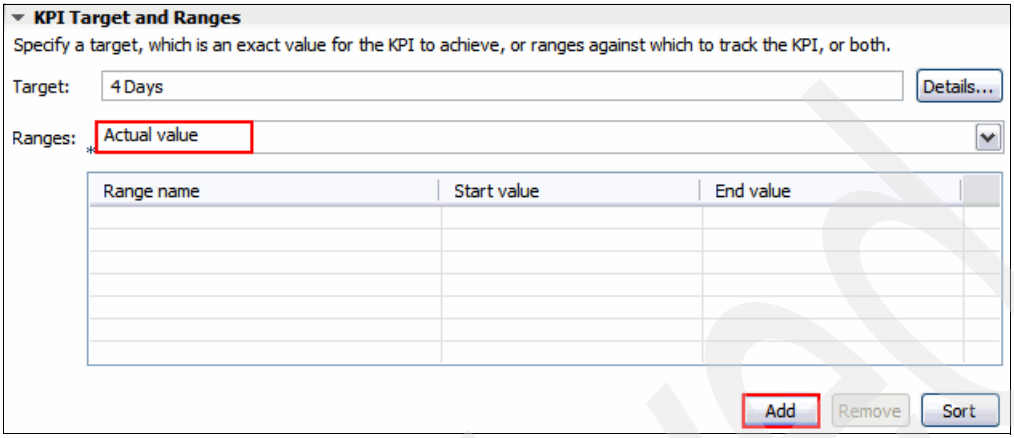


Figure 7-96 KPI Target and Ranges

   b. In the Add Range window, for Name, type `Good` and click **OK**.
   c. For the Good range (Figure 7-97), leave the Start Value as 0 Milliseconds, and click the button in the End value field.

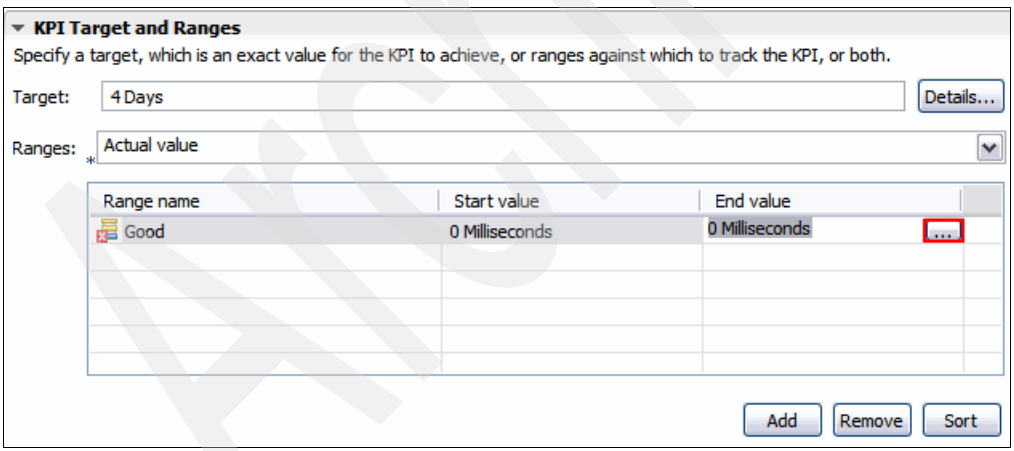


Figure 7-97 KPI Target and Ranges - Good range

d. In the Specify End Value window (Figure 7-98), for Days Value, type 3 and click **OK**.

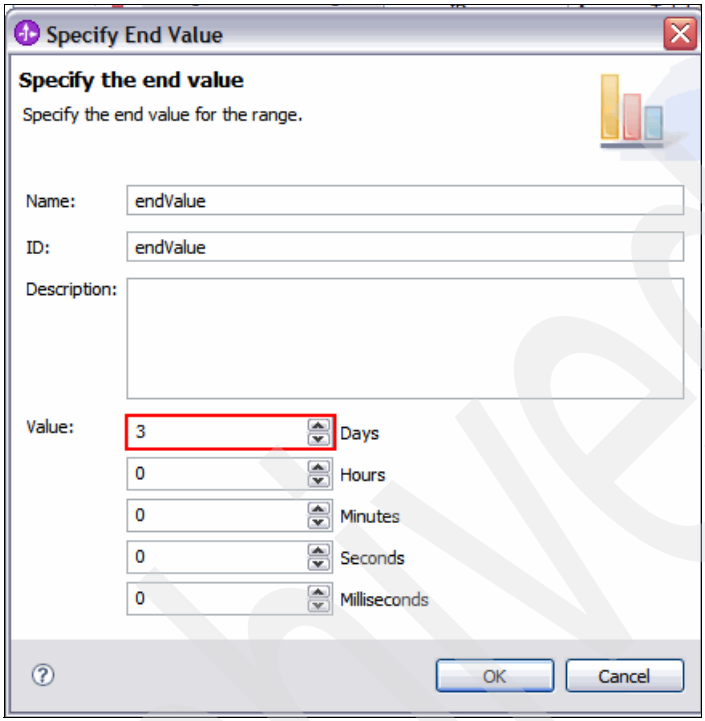


Figure 7-98 Specify End Value window

e. Click **Add** to add another range and add a range of `Average` starting at `3 Days` and ending at `<5 Days`.

f. Click **Add** to add another range and add a range of `Poor` starting at `5 days` and ending at `<7 days`.

9. Specify the base metric that this KPI is based on. This KPI is based on a metric and aggregation function. In the KPI Definition section under KPI Value (Figure 7-99), select **Base this KPI on a metric and an aggregation**.

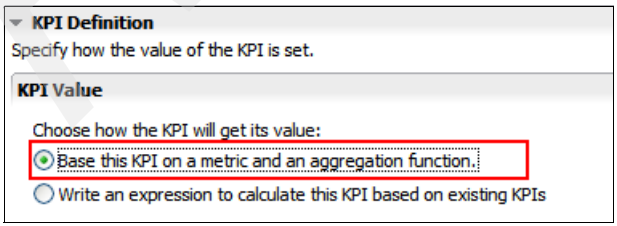


Figure 7-99 KPI Value

10. Under KPI Details (Figure 7-100):

    a. For Monitoring context, click **Browse**. Expand **P8 BPM Monitoring** and select **P8 BPM Process MC**. Click **OK**.

    b. For Metric, click **Browse**. Expand **P8 BPM Process MC** and select **Processing Time Stopwatch**. Click **OK**.

    c. For Aggregation function, select **Average**.

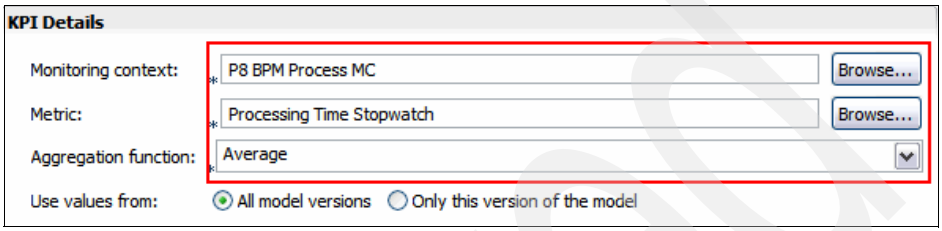


Figure 7-100 KPI Details

Creating the Average Total Processing Time Indexing Queue KPI

To create an Average Total Processing Time Indexing Queue KPI:

1. In the KPI Context navigator, right-click **Average Total Processing Time** and select **New** → **KPI**.
2. In the Create New KPI window, for Name, type `Average Total Processing Time Indexing Queue` and click **OK**.
3. In the KPI Details section, for Type, select **Duration**.
4. In the KPI Target and Ranges section, next to the Target field, click the **Details** button.
5. In the Target Details window, for Days Value, type `1` and click **Finish**.
6. Enter the ranges:

    a. For Range type, keep the **Actual value** option. Click **Add** to add a range.

    b. In the Add Range window, for Name, type `Good` and click **OK**.

    c. Leave the Start Value as `0` Milliseconds. Click the button in the End value field.

    d. In the Specify End Value window, for Days Value, type `1` and click **OK**.

    e. Click **Add** to add another range. Add a range of `Average` starting at `1 Days` and ending at <`2 Days`.

    f. Click **Add** to add another range. Add a range of `Poor` starting at `2 days` and ending at <`3 days`.

Archived

7. This KPI is based upon a metric and an aggregation function. Specify the base metric that this KPI is based upon:
   a. In the KPI Definition section under KPI Value, select **Base this KPI on a metric and an aggregation**
   b. Under KPI Details, for Monitoring context, click **Browse** and expand **P8 BPM Monitoring** → **P8 BPM Process MC**. Select **P8 BPM Work Item MC**. Click **OK**.
   c. For Metric, click **Browse** and expand **P8 BPM Process MC** → **P8 BPM Work Item MC**. Select **Total Indexing Queue Processing Time Stopwatch**. Click **OK**.
8. From the Aggregation function list, select **Average**.
9. Under Data Filter (Figure 7-101):
   a. Click **Add**.
   b. Browse to **P8 BPM Work Item MC** and select **Total Indexing Queue Processing Time Stopwatch** and click **OK**.
   c. In the Operator cell, type `greaterThan`.
   d. In the Values cell, type `xs:dayTimeDuration('P0D')`.

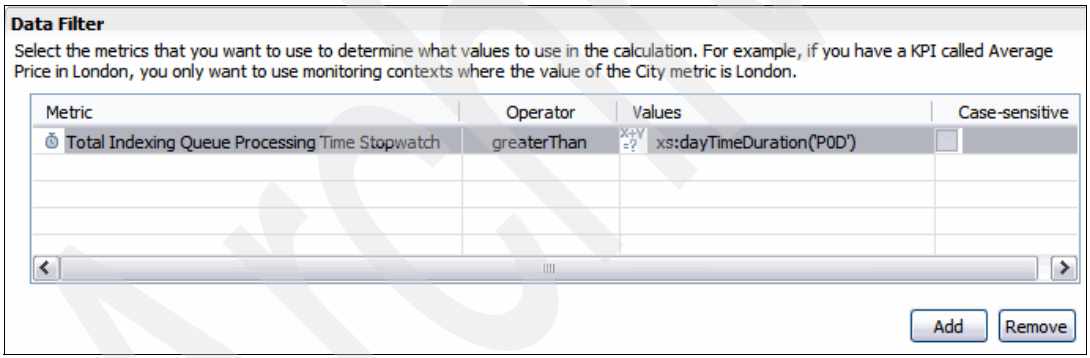


Figure 7-101 Data Filter

10. Press Ctrl+S to save your work.

Creating other KPIs

Repeat the instructions in the previous section to create KPIs for the other activities by using the Metrics and Data Filters as described in Table 7-15 on page 445. The target and ranges of these KPIs are same as Average Total Processing Time Indexing Queue KPI.

Table 7-15 Metrics

KPI name	Metric/data filter
Average Total Processing Time Pended Index Queue	Total Pended Index Queue Processing Time Stopwatch
Average Total Processing Time Review Queue	Total Review Queue Processing Time Stopwatch
Average Total Processing Time Pended Review Queue	Total Pended Review Queue Processing Time Stopwatch
Average Total Processing Time Approve Queue	Total Approve Queue Processing Time Stopwatch

When you are finished, press Ctrl+S to save your work.

7.9.10 Defining the visual model

In this section, we explain how to attach an SVG diagram to a context. You associate this SVG file with the KPIs that were created previously to display the KPIs value in the visual model.

SVG file: We provide the SVG file Contract_Management_Process.svg that is used in this section in the additional materials for this book. This SVG file is the Contract Management Process exported from WebSphere Business Modeler. For details, see Appendix B, "Additional material" on page 801.

Preparing the SVG diagram

To prepare the SVG diagram:

1. In the Project Explorer view, right-click **BPMMonitoring** and select **Import** → **General** → **File system**. Click **Next**.
2. For From directory, click **Browse**. Go to the directory that contains the SVG file and click **OK**. Select the SVG file **Contract_Management_Process.svg**.
3. For Into folder, click **Browse** and locate the monitor model project **BPMMonitoring**. Click **Finish**.

   The SVG file is placed in the SVG Files group in the Project Explorer view.
4. In the editor, click the **Visual Model** tab, and for Selected Context, select **Average Total Processing Time**.

Archived

5. In the Visual Model editor pane (Figure 7-102), click **Browse**. Locate the SVG file **Contract_Management_Process.svg** and click **OK**.

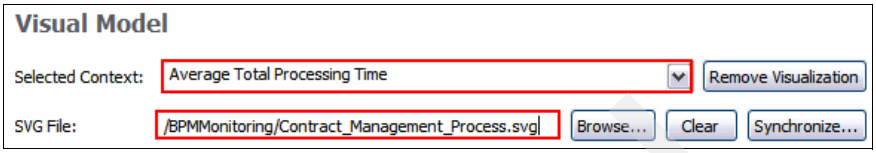


Figure 7-102 Visual Model pane

6. If you are prompted to create shape sets based on the SVG file, click **Yes**. Shape sets are created based on any shapes in the file that have monitor model IDs.

The diagram is now associated with a context. When the context is chosen in the Diagram view in the WebSphere Business Monitor dashboards, the diagram is displayed (Figure 7-103).

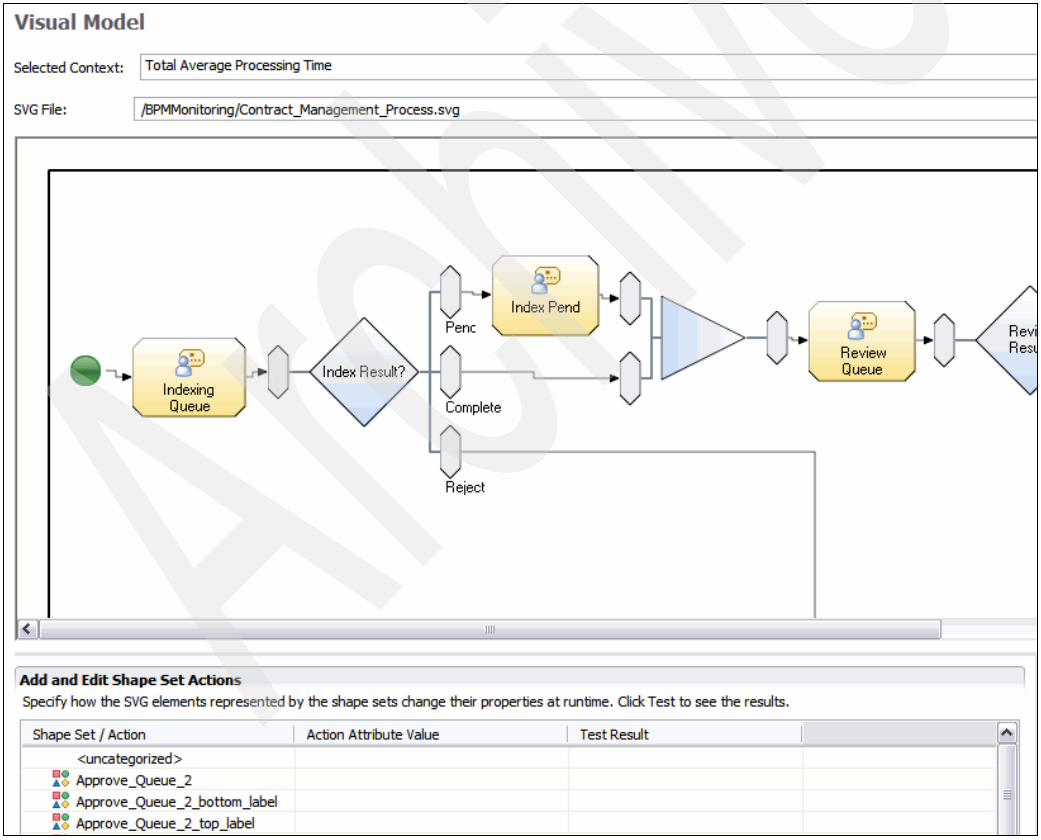


Figure 7-103 Contract Management diagram

Archive

7.9.11 Defining actions

To be updated dynamically at run time, an SVG file needs a group of named shape sets for referencing sets of individual shapes in the SVG diagram. It also needs a set of actions, such as color or text changes, that describe how the SVG elements represented by the shape sets change their properties based on the values of metrics or KPIs at run time. We define actions to update the Average Total Processing Time KPIs of each activity.

Defining actions for the Indexing Queue

To define the actions for the Indexing Queue:

1. Place your cursor on the diagram and find the shape set in the area of **Indexing Queue**. The cursor changes to a hand when it is on the shape set. Click the mouse to highlight the shape set in the table at the bottom of the page (Figure 7-104).

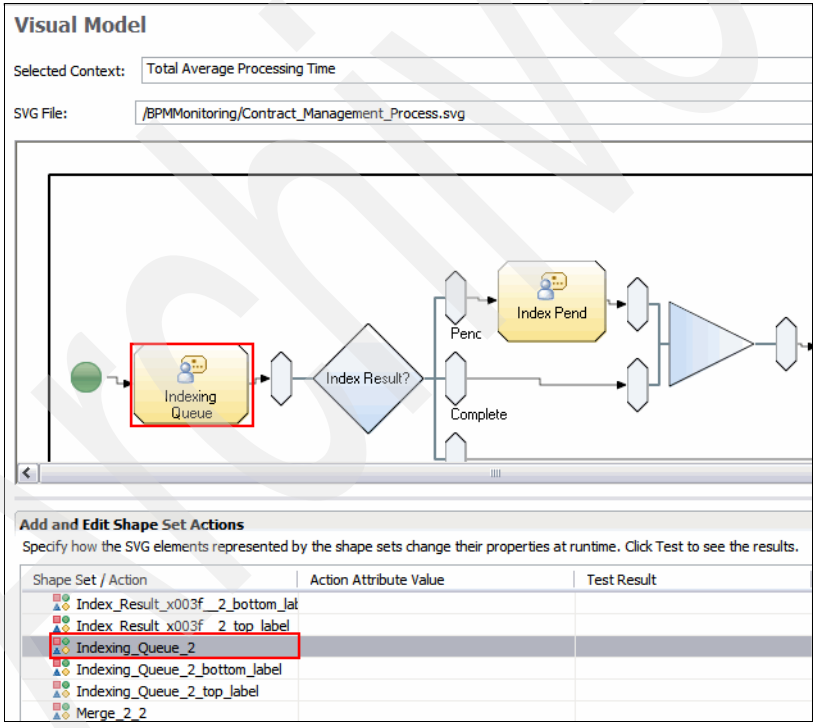


Figure 7-104 Selecting the Indexing Queue

Archived

2. Right-click the **Indexing_Queue_2** shape set in the table and click **Add Action** → **Set Color** (Figure 7-105).

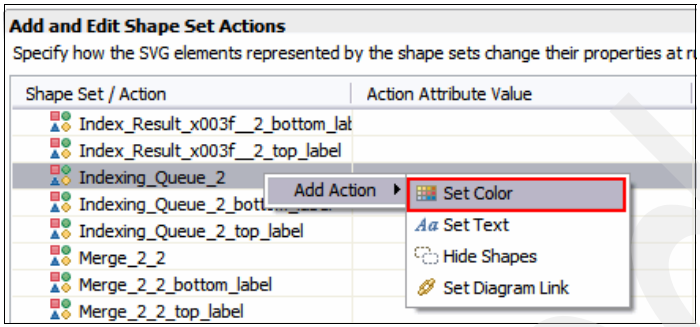


Figure 7-105 Selecting the option to add and edit the shape set actions

3. Under Add and Edit Shape Set Actions (Figure 7-106):
   a. In the condition field, type `Average_Total_Processing_Time_Indexing_Queue gt xs:dayTimeDuration('P1D')`.
   b. Click the button in the fill color row.
   c. In the color window that opens, choose a color or click **Define Custom Colors** to create your own. In this example we use `#ff8080`.

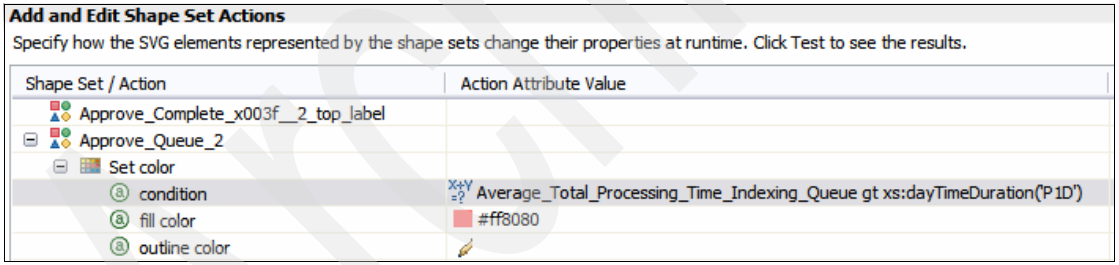


Figure 7-106 Add and Edit Shape Set Actions section - Setting the color to #ff8080

   d. Right-click the **Indexing_Queue_2** shape set in the table and click **Add Action** → **Set Color**.
   e. In the condition field, type `Average_Total_Processing_Time_Indexing_Queue le xs:dayTimeDuration('P1D')`.
   f. Click the button in the fill color row.
   g. In the color window that opens, choose a color or click **Define Custom Colors** to create your own. In this example, we use `#80ff80`.

Archived

Figure 7-107 shows the changes from steps e through g.

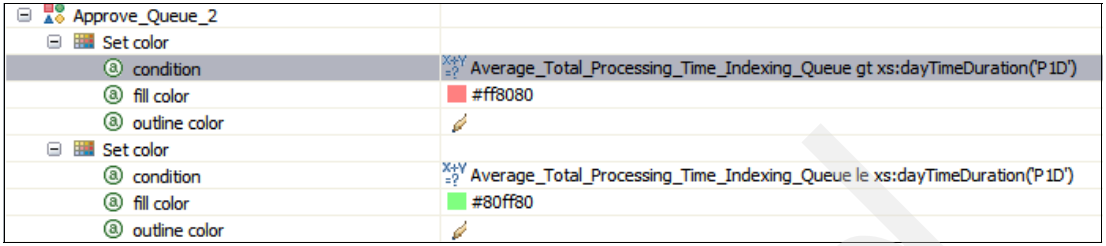


Figure 7-107 Setting the color to #80ff80

h. Right-click the **Indexing_Queue_2_top_label** shape set in the table and click **Add Action** → **Set Text**.

i. In the text value field, type `Average_Total_Processing_Time_Indexing_Queue` (Figure 7-108).

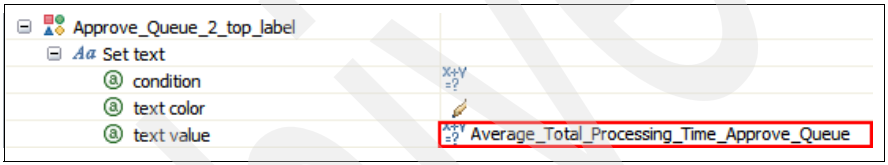


Figure 7-108 Setting the text to Average_Total_Processing_Time_Indexing_Queue

Defining actions for other activities

Repeat the steps in the previous section to create actions for the following activities:

- Average Total Processing Time Pended Index Queue
- Average Total Processing Time Review Queue
- Average Total Processing Time Pended Review Queue
- Average Total Processing Time Approve Queue

Archived

7.9.12 Creating situation events

In this section, we explain how to create a situation event if the contract fulfillment time is greater than four days.

Creating the outbound event

To create the outbound event for a contract processing time situation:

1. Create the event definition ContractFulfillmentTime.cbe:
   a. In the Project Explorer view, right-click **Event Definitions** and select **New → Event Definition (cbe)**.
   b. For file name, enter `ContractFulfillmentTime.cbe`, and click **Finish**.
   c. In the Event Definition Editor, under the Extended Data section (Figure 7-109):
      i. Click the icon to add an extended data element.
      ii. Change Name to `BusinessSituationName` and change Type to **string**.
      iii. Click the icon to add an extended data element.
      iv. Change Name to `ContractDuration` and change Type to **string**.

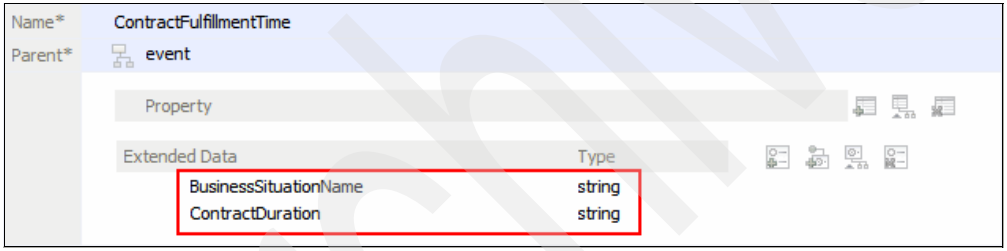


Figure 7-109 ContractFulfillmentTime event definition

2. Press Ctrl+S to save your work.
3. Create the Average Processing Duration Trigger 1 trigger. This trigger is used to fire the situation event when the process duration is too long (more than four days). In the KPI Model panel:
   a. In the BPM Monitoring model, click the **KPI Model** tab.
   b. Right-click in the navigator, select **New → Trigger**.
   c. For Name, type `Average Processing Duration Trigger 1` and click **OK**.
   d. For Trigger Source, click the **Add** button.

Archived

e. In the Select Trigger Source window (Figure 7-110), select **Recurring wait time** and click **OK**.

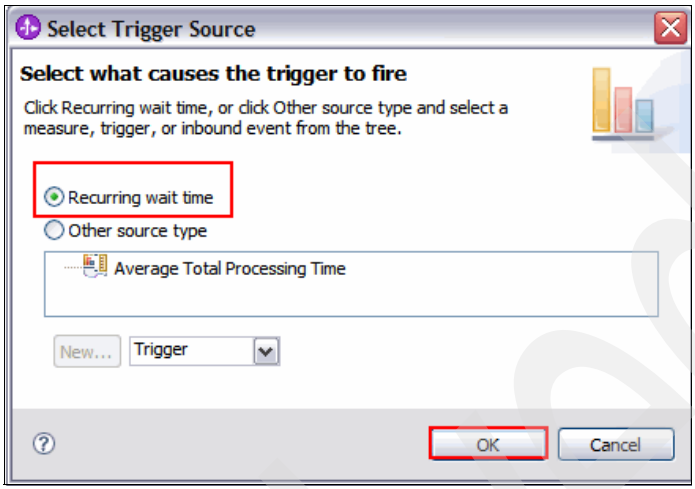


Figure 7-110 Select Trigger Source window

4. Under Trigger condition (Figure 7-111), type `Average_Total_Processing_Time ge xs:dayTimeDuration('P4D')`.

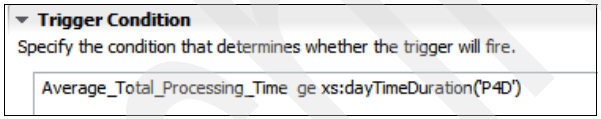


Figure 7-111 Trigger Condition

5. Create the outbound event Contract Fulfillment Event:
   a. In the Monitor Model editor, click the **KPI Model** tab.
   b. In the navigator, right-click and select **New** → **Outbound Event**.

Archived

c. In the Create New Outbound Event window (Figure 7-112):

   i. For Name, type `Contract Fulfillment Event`.

   ii. Select the **Configure this event to generate an alert in the dashboards** check box.

   iii. For Trigger, click **Browse**. Expand **Average Total Processing Time** and select **Average Processing Duration Trigger 1**. Then click **OK**.

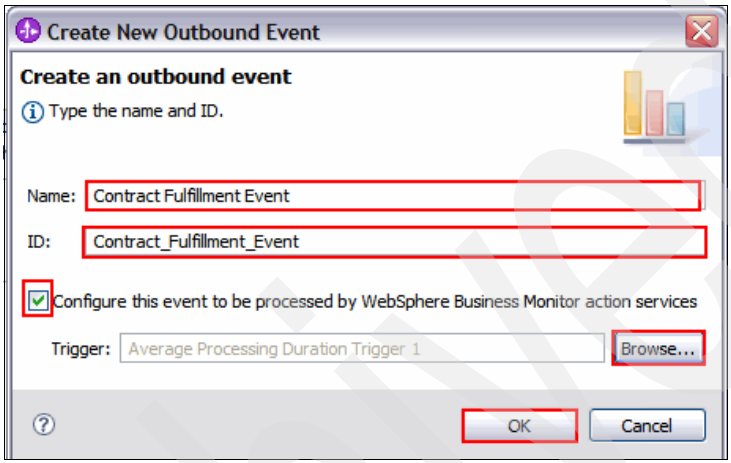


Figure 7-112 Create New Outbound Event window

6. Under Event Type Details (Figure 7-113), for Extension name, click **Browse**. Expand **BPMMonitoring** → **ContractFulfillmentTime.cbe** and select **ContractFulfillmentTime**. Click **OK**.

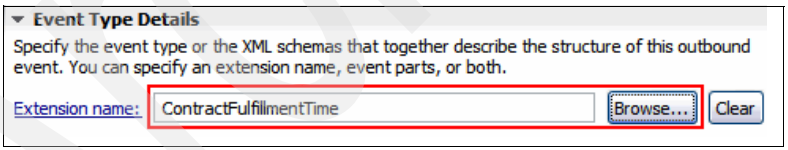


Figure 7-113 Event Type Details

7. Under Event Attributes Details (Figure 7-114):

   a. Navigate to **BusinessSituationName**. You see that the value of the situation name is set to `Contract Fulfillment Event`.

   b. Set the expression of ContractDuration to `xs:string(Average_Total_Processing_Time)`.

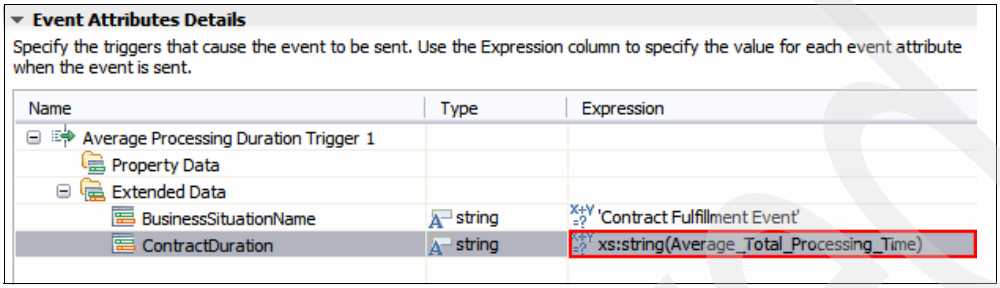


Figure 7-114 Event Attributes Details

8. Press Ctrl+S to save your work.

7.10 Generating and deploying the monitor model

In this section, we explain how to perform the following tasks:

- Generating the Monitor J2EE projects
- Generating the EAR file
- Deploying the monitor model

Archived

7.10.1 Generating the Monitor J2EE projects

After you finish the monitor model, generate a J2EE project from the monitor model:

1. In Project Explorer, expand **BPMMonitoring** → **Monitor Model**. Right-click **BPMMonitoring.mm** and select **Generate Monitor J2EE Projects** (Figure 7-115).

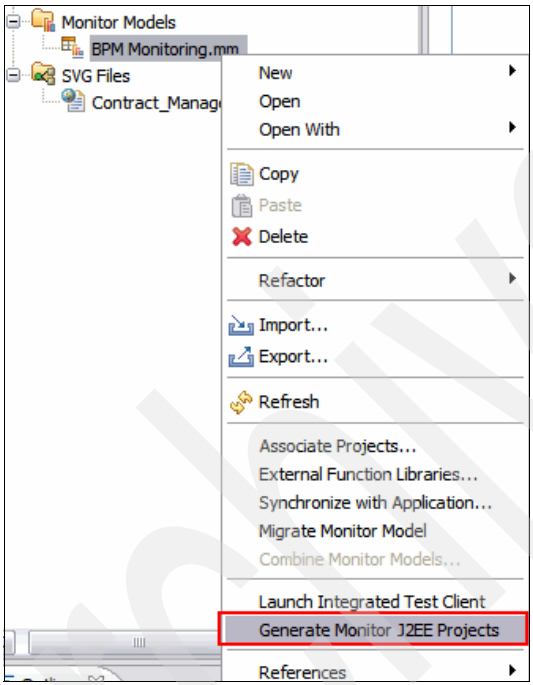


Figure 7-115 Selecting Generate Monitor J2EE Projects

Archived

2. In the Generate Monitor J2EE Projects window (Figure 7-116), accept the name defaults and then click **Finish**.

Notes:

- You can accept the default names for the projects, but be sure that the names are not too long. In Windows, there is a maximum path length of 256 characters.
- Since this is the first time we are generating the projects, we do not need to select the *overwrite existing projects* box.

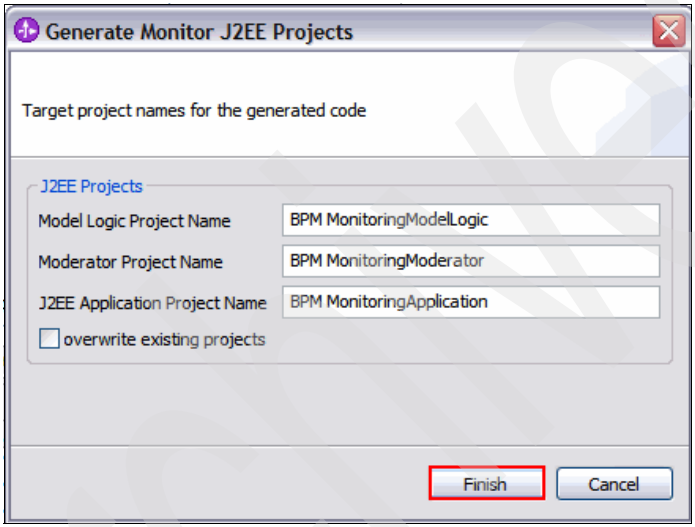


Figure 7-116 Generate Monitor J2EE Projects window

Archive

A progress bar is displayed during the generation and closes automatically when the generation is complete. It might take some time to complete.

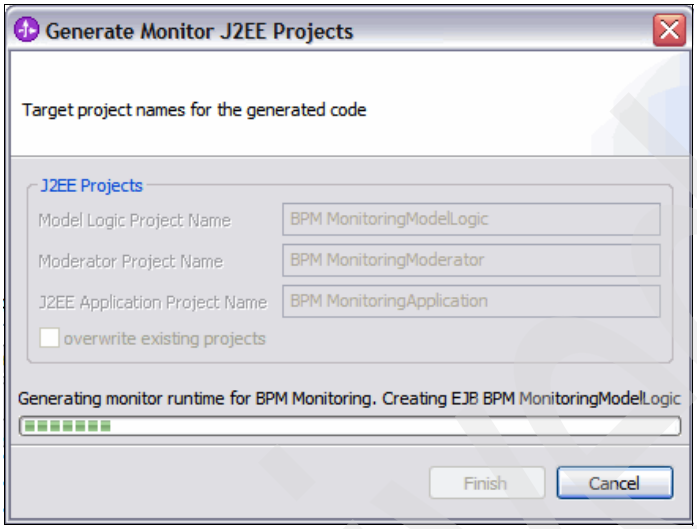


Figure 7-117 Generate Monitor J2EE Projects

3. After the processing completes, check for error messages in the Problems view.

Note: The following three J2EE projects are generated from the monitor model:

- A Model Logic EJB project that contains the logic of the monitor model
- A Moderator EJB project that receives events from the business applications and routes them to the target monitor model
- An enterprise application project that contains the two EJB projects

7.10.2 Generating the EAR file

To create the EAR file of the Monitor J2EE Project in order to deploy it by using the WebSphere Administrative Console:

1. Switch to the J2EE perspective. Click the **Open perspective** shortcut, select **Other** → **J2EE**, and then click **OK**.
2. Right-click **BPMMonitoringApplication** and select **Export** → **EAR file** (Figure 7-118 on page 457).

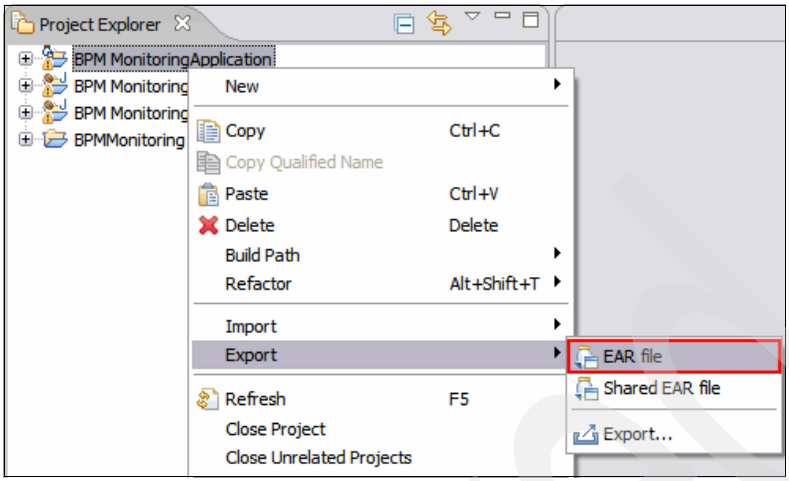


Figure 7-118 Selecting to export the EAR file

3. In the Export window (Figure 7-119), click the **Browse** button to select a destination, for example the C:\EARS directory. Then click **Finish**.

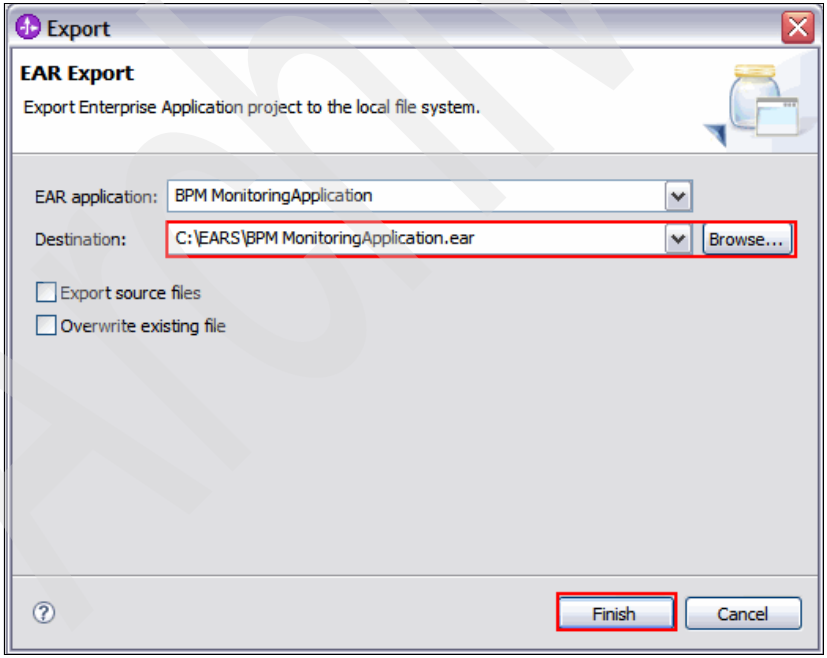


Figure 7-119 Exporting the EAR file

You now have an EAR file in the target directory.

7.10.3 Deploying the monitor model

To deploy the EAR file that we generate in the previous section, see 3.4.1, “Installing a monitor model by using the default settings” on page 91.

To start the monitor model, see 3.4.3, “Starting and stopping a monitor model” on page 103.

7.11 Configuring action manager for business situation events

In this section, we show you how to configure the action manager to create alerts based on business situation events defined in the monitor model. The alerts are visible in the Alert view on your dashboard. To configure the action manager:

1. Add a template for contract processing time alerts:
   a. In the WebSphere Administrative Console, expand **Application →Monitor Action Service →Template Definition** and click **Notifications**.
   b. In the right pane, click **New**.
   c. In the Notifications panel (Figure 7-120 on page 459):
      i. For Template name, type `AlertTime` and enter a description.
      ii. Under Default action service type, select **Dashboard Alert**.
      iii. Under 'To' query type, select **User id**.
      iv. For To, type `admin`. This user ID receives the alert. Therefore, you must log into the dashboard with this user ID.
      v. For Subject, type `Contract Processing Time`.
      vi. For Body, type a description. You can optionally enter a substitution variable. For the actual alert that is sent, the average contract processing time from the event payload is substituted for %ContractDuration% in the body, for example: `The average contract processing time is %ContractDuration% days`.
      vii. Click **OK**.

Archived

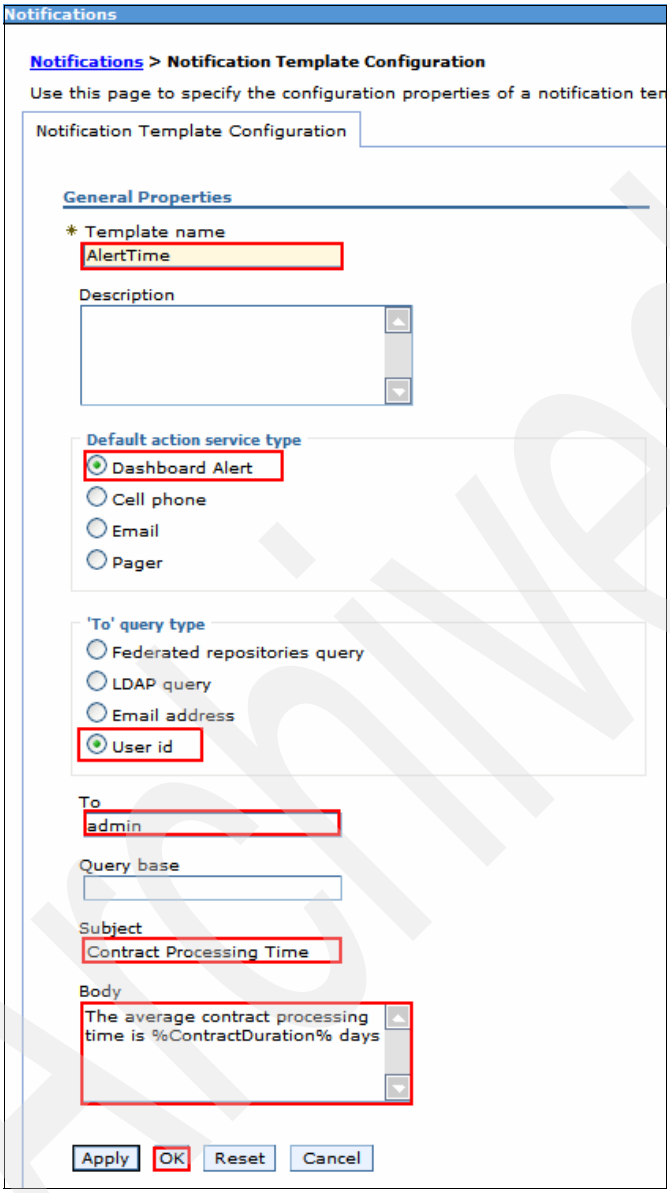


Figure 7-120 Notifications

Archived

2. Add the binding from the situation event to the action type for contract processing time:

   a. In the WebSphere Administrative Console, expand **Application** → **Monitor Action Service** and click **Installed Situation Event Bindings**.

   b. Click **New**.

   c. Under General Properties (Figure 7-121), for Situation event name, type `Contract Fulfillment Event` and type a description. Then click **Apply**.

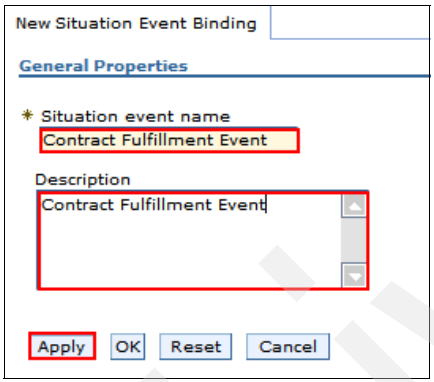


Figure 7-121 New Situation Event Binding

3. Click **Add**.

4. Under General Properties (Figure 7-122), for Binding name, type `Contract Fulfillment Event`. For template name, select **AlertTime**. Then click **OK**.

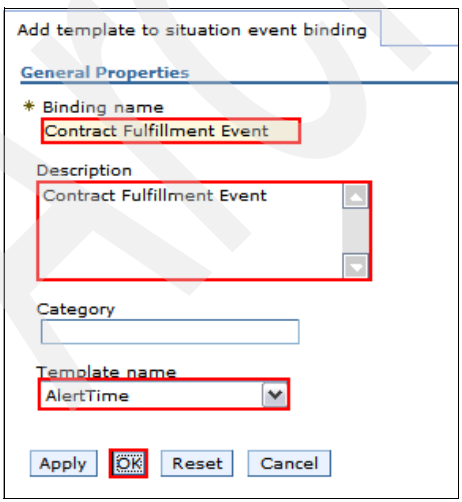


Figure 7-122 Add a template to situation event binding page

Archived

5. Click **OK** again.

Figure 7-123 shows that the template and binding name have been added.

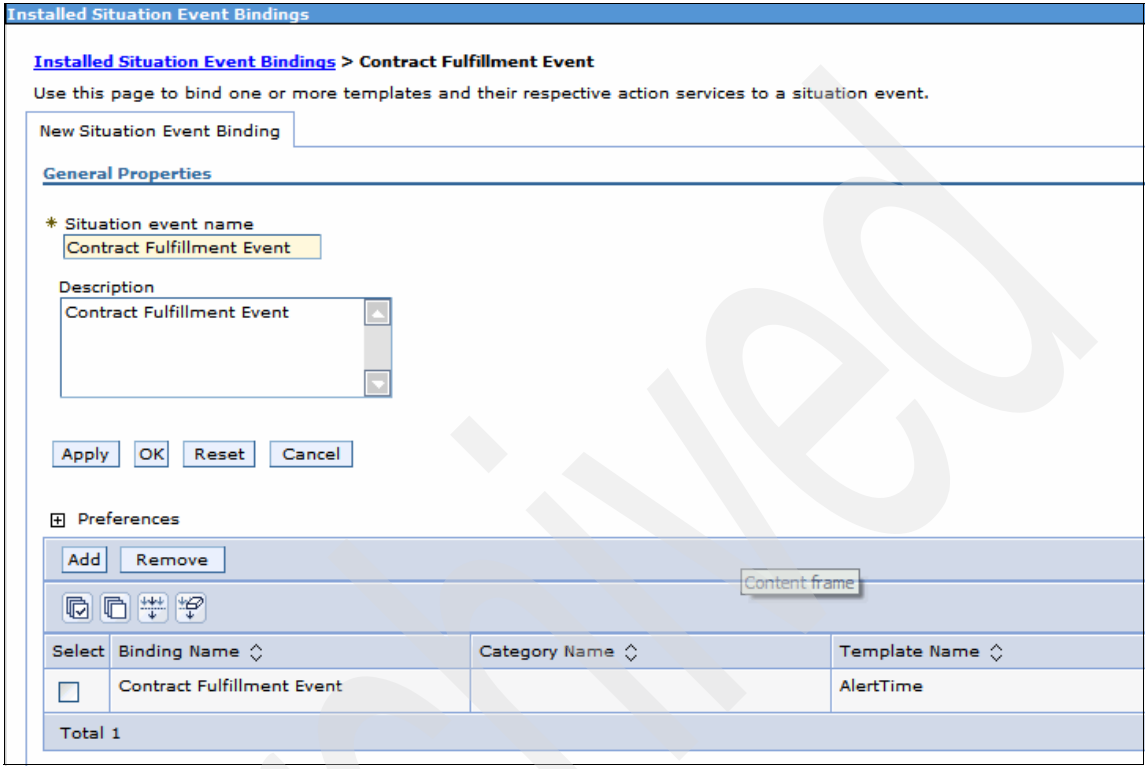


Figure 7-123 Installed situation event bindings

Archived

7.12 Testing the monitor model

To run the Contract Management Workshop, see 7.8, “Running the Contract Management Workflow” on page 404.

7.13 Creating a dashboard

In this section, you build a dashboard, add views to the dashboard, and configure them:

1. Open the dashboard:
   a. To launch the WebSphere Business Monitor dashboard, open a browser and type the following URL:

      `http://[WebSphere Business Monitor host]:9080/BusinessDashboard`

   b. Log in as `admin`.
   c. Click the **Dashboards** tab. On the Manage tab, click **New** (Figure 7-124).

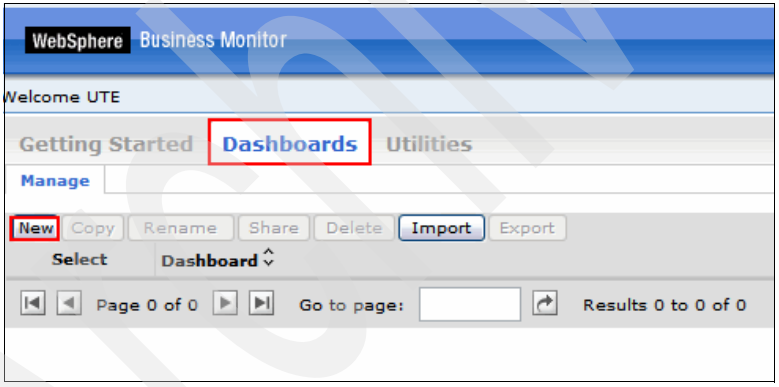


Figure 7-124 Dashboard

Archived

2. In the New Dashboard panel (Figure 7-125), for Name, type `Contract Management` and click **OK**.

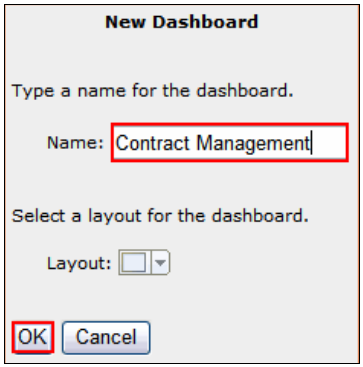


Figure 7-125 New Dashboard panel

3. Add the Instances view to see the monitored instances:
   a. Click **Add to Dashboard**.
   b. In the Add to Dashboard panel (Figure 7-126), select **Instances** and click **OK**.

> **Note:** You can also add a view by dragging the view from the palette on the right to the dashboard.

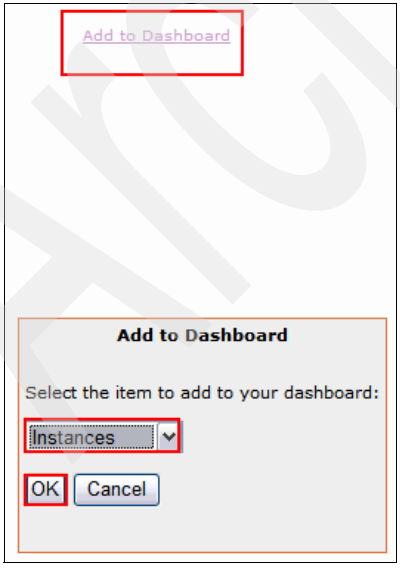


Figure 7-126 Add to Dashboard

Archived

c. Under Instances (Figure 7-127), click **Personalize**.

Figure 7-127 Instances

d. Click the **Advanced** tab (Figure 7-128). For Model, select **P8 BPM Monitoring (All Versions)**.

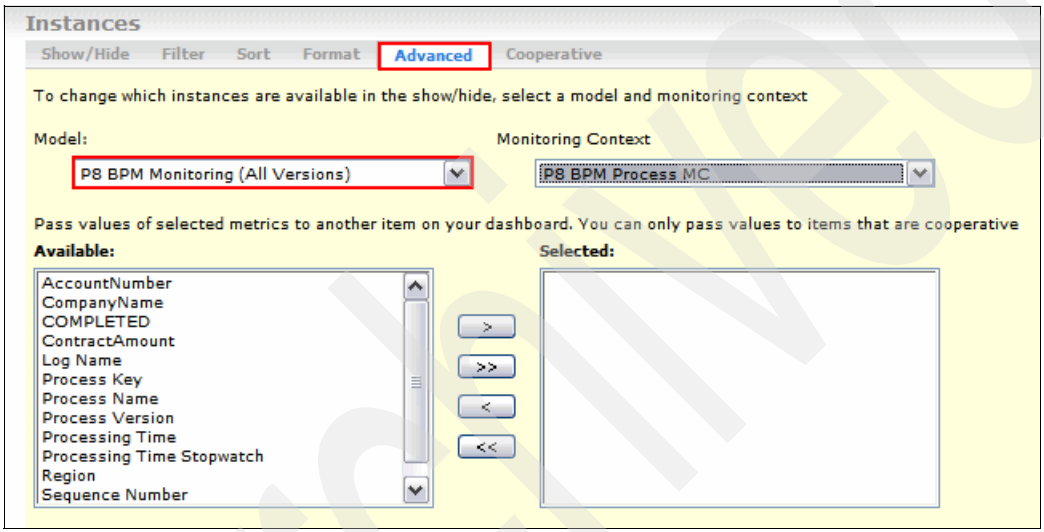


Figure 7-128 Advanced tab

Archived

e. Click the **Show/Hide** tab (Figure 7-129).

i. In the Available list, select **AccountNumber**, **CompanyName**, **ContractAmount** and **Processing Time Stopwatch**.

ii. Click **>** to copy the these values to the selected list.

iii. Click **OK**.

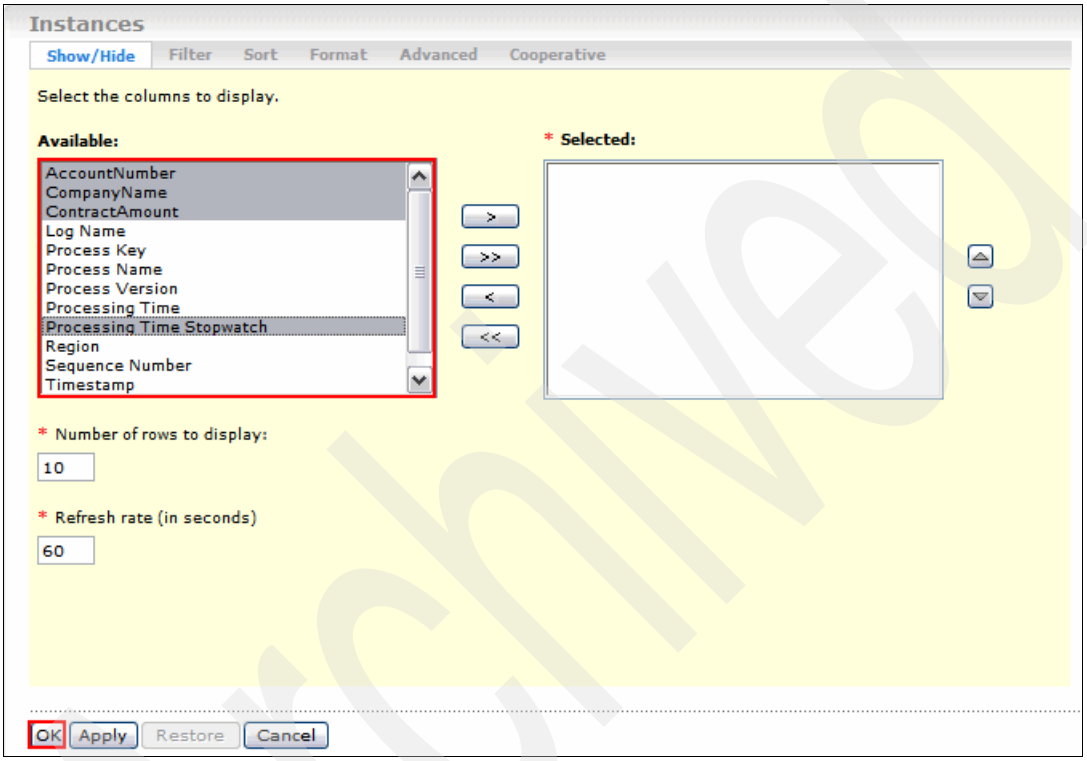


Figure 7-129 Show/Hide

Archived

You see a list of monitoring context instances for the events that you just processed (Figure 7-130).

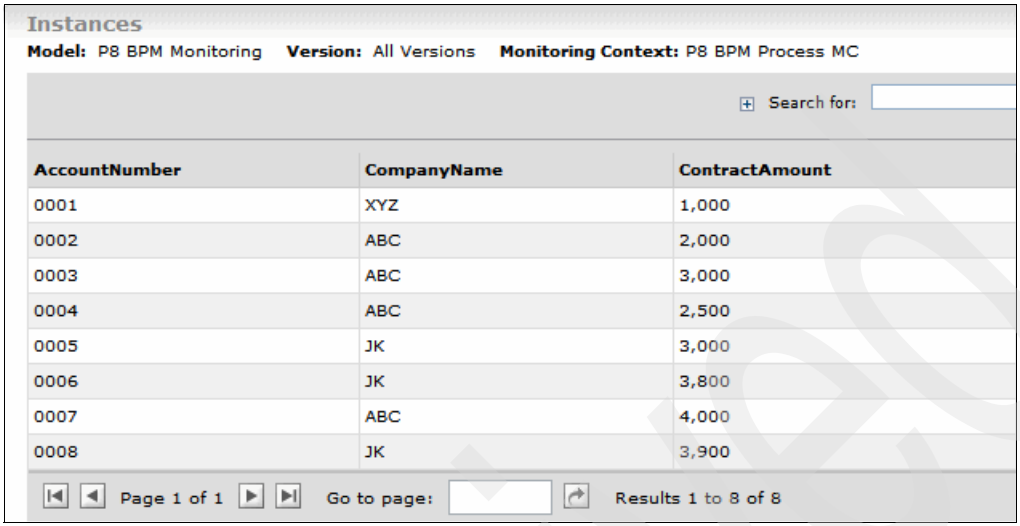


Figure 7-130 Instances

4. Add the Dimensions view to see process dimensions:
   a. Click **Add to Dashboard**.
   b. On the Add to Dashboard page, select **Dimensions** and click **OK**.

Archived

c. Under Dimensions (Figure 7-131):

i. Click **Personalize**.

ii. For Monitoring Model, select **P8 BPM Monitoring (All Versions)** and for Monitoring context, select **P8 BPM Process MC**.

iii. For Available dimensions, select **Company** and click **>** next to Column dimensions. Select **Measures** and click **>** next to Page dimensions.

iv. Click **OK**.

TerminationTime: The FileNet P8 BPM Monitor Model for WebSphere is intended to monitor work in progress. To see the number of active processes, you must use *TerminationTime* as a dimension and show only data based on TerminationTime=9999.

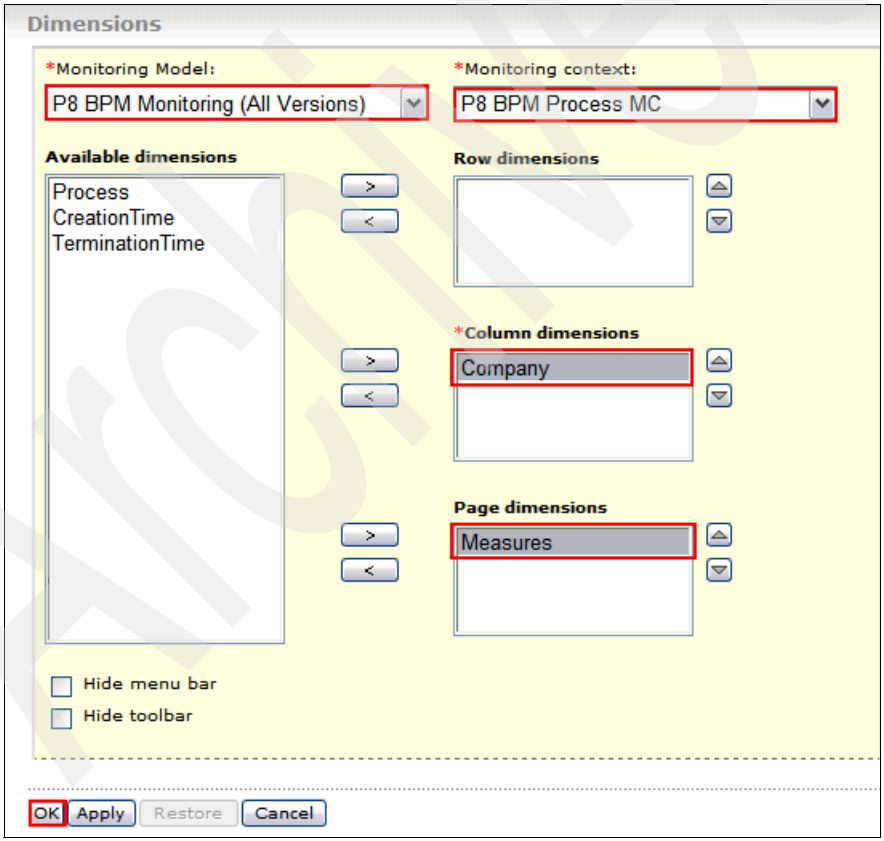


Figure 7-131 Dimensions view

You see a Company dimension (Figure 7-132).

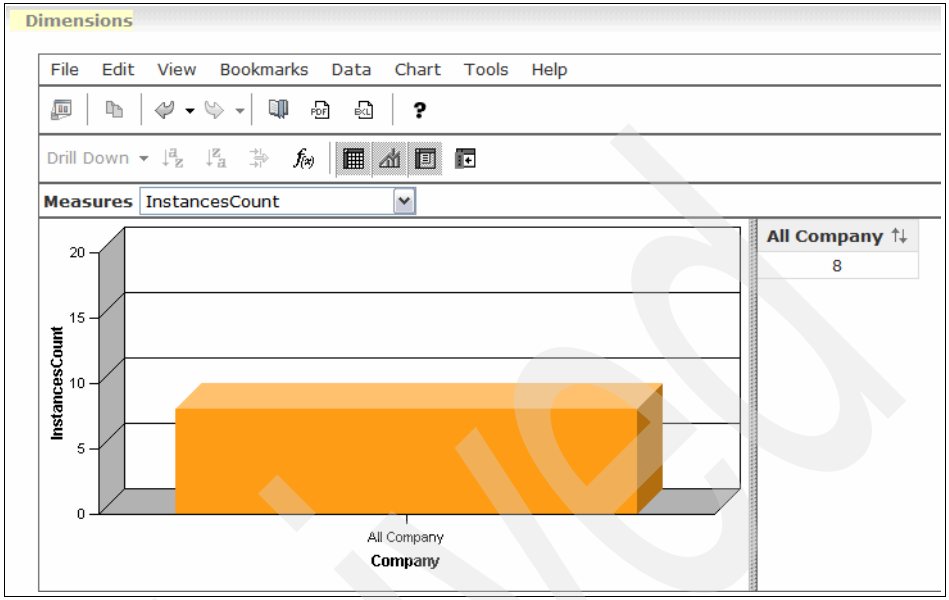


Figure 7-132 Company dimension

Archived

5. Right-click the **All Company** bar and select **Drill Down** (Figure 7-133).

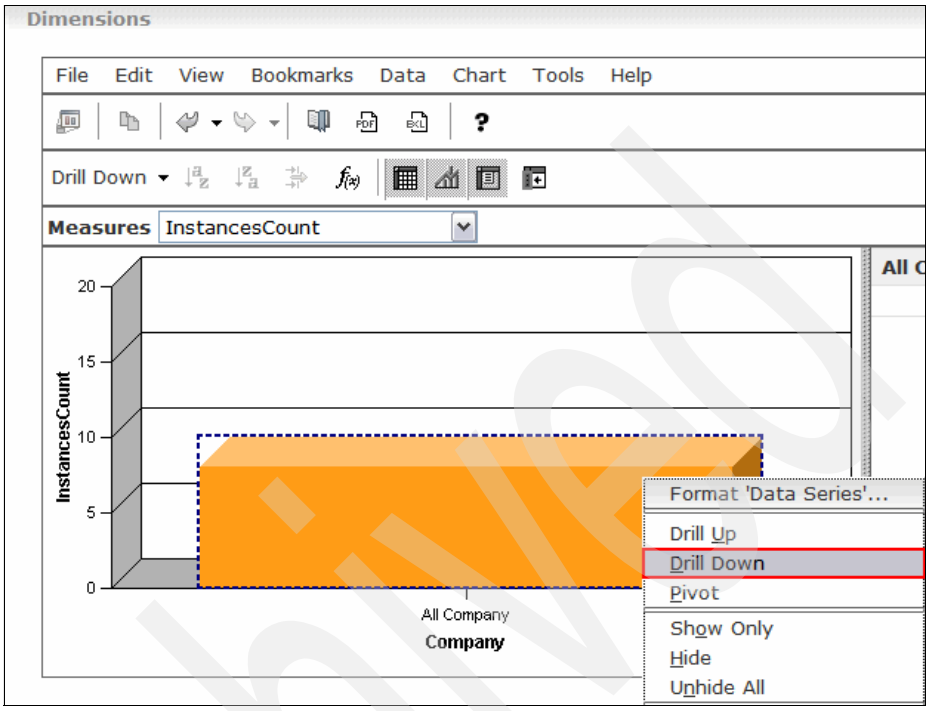


Figure 7-133 Selecting Drill Down

Archived

You see a graph that represents the number of contracts by company (Figure 7-134).

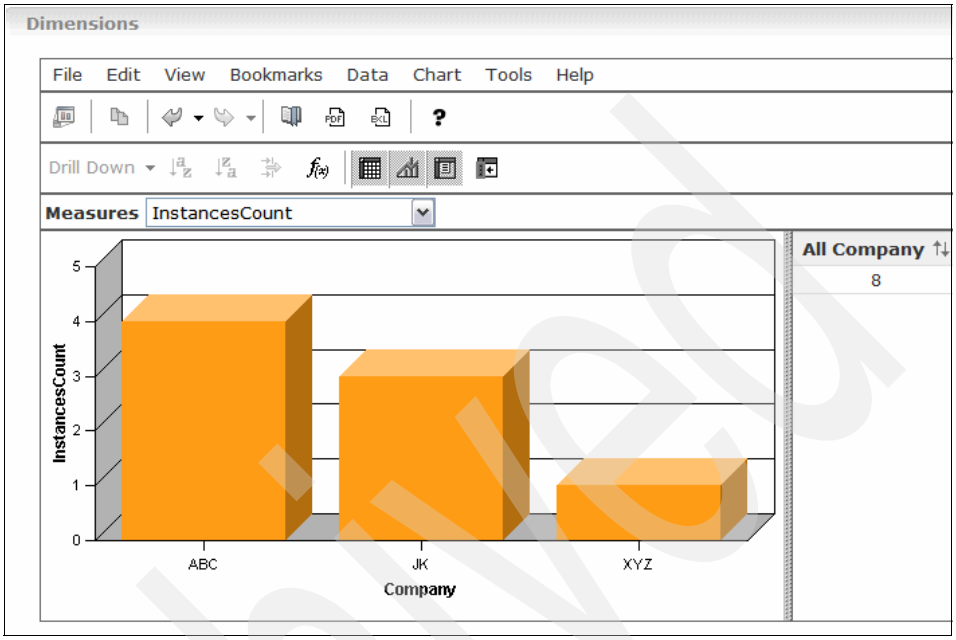


Figure 7-134 Drill Down Company dimension

Archived

6. Add the Dimensions view to see activity dimensions:
   a. Click **Add to Dashboard**.
   b. On the Add to Dashboard page, select **Dimensions** and click **OK**.
   c. Under Dimensions (Figure 7-135):
      i. Click **Personalize**.
      ii. For Monitoring Model, select **P8 BPM Monitoring (All Versions)** and for Monitoring context, select **P8 BPM Work Item MC**.
      iii. For Available dimensions, select **Task** and click **>** next to Column dimensions. Select **Measures** and click **>** next to Page dimension.
      iv. Click **OK**.

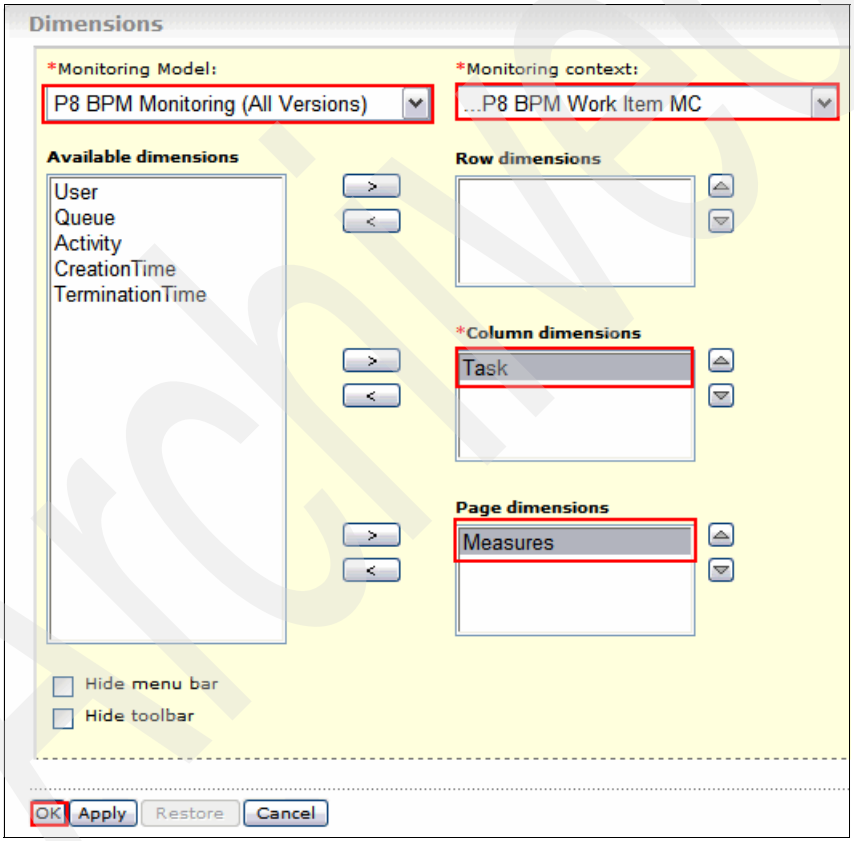


Figure 7-135 Dimensions view

Archive

The Task dimension is displayed (Figure 7-136).

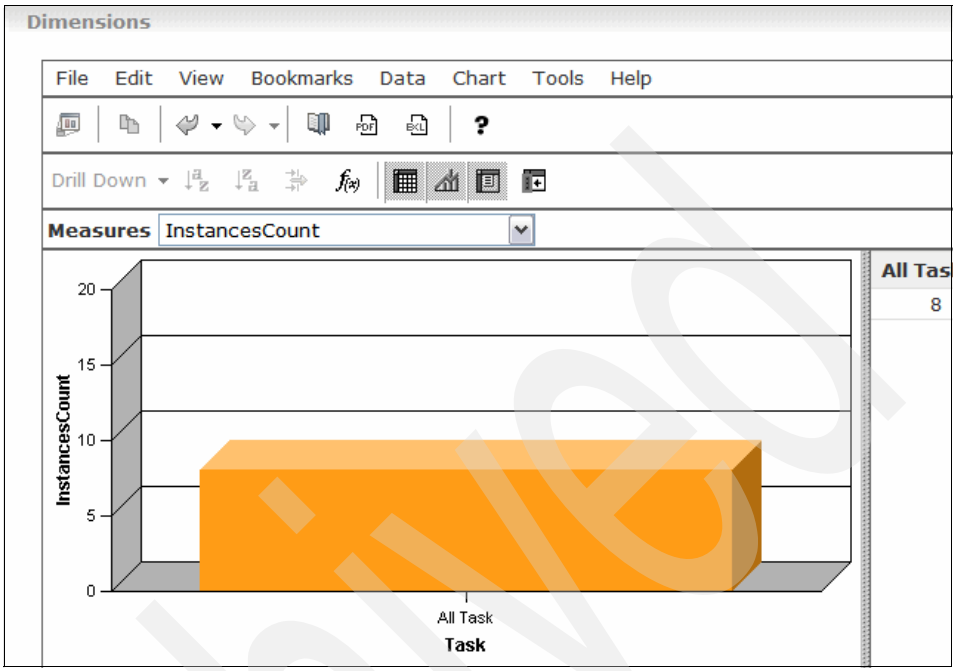


Figure 7-136 Task Dimension

7. Right-click the **All Task** bar and select **Drill Down**.

Archived

You see a graph that represents the number of work items by map (Figure 7-137).

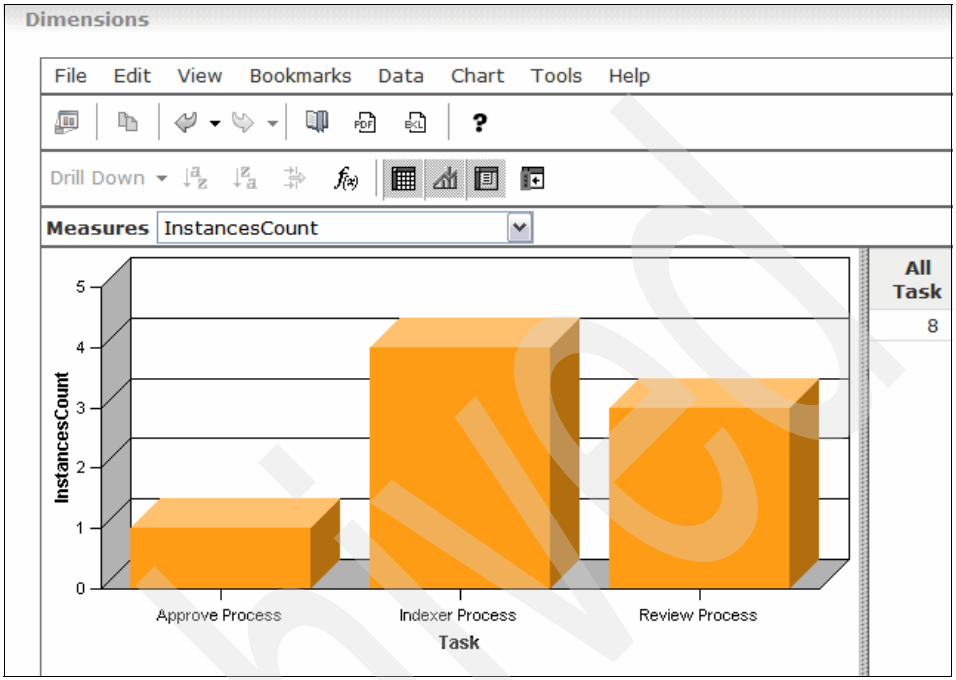


Figure 7-137 Drill Down Task dimension

8. Right-click the **Index Process** bar and select **Drill Down**.

Archive

You see a graph that represents the number of work items by activity (Figure 7-138).

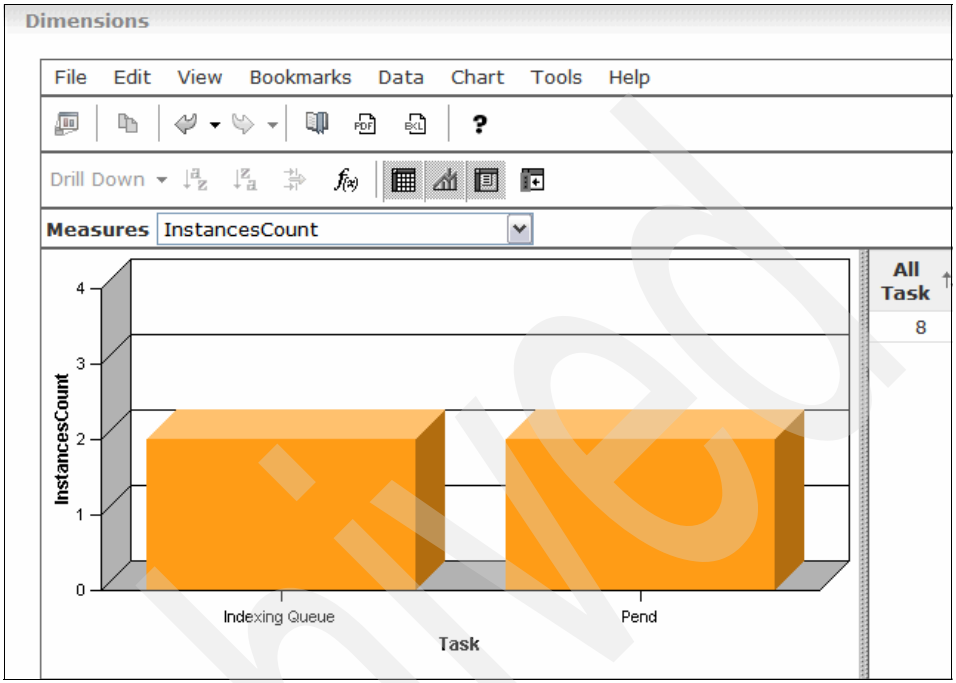


Figure 7-138 Drill Down Indexer Process

9. Add the KPIs view to see KPIs of Contract Management Workflow:
   a. Click **Add to Dashboard**.
   b. On the Add to Dashboard page, select **KPIs** and click **OK**.
   c. Click **Personalize**.

Archived

d. On the Select KPIs page (Figure 7-139), for Model version, select **All Versions** and then select **P8_BPM_Monitoring <Version>**. Click **OK**.

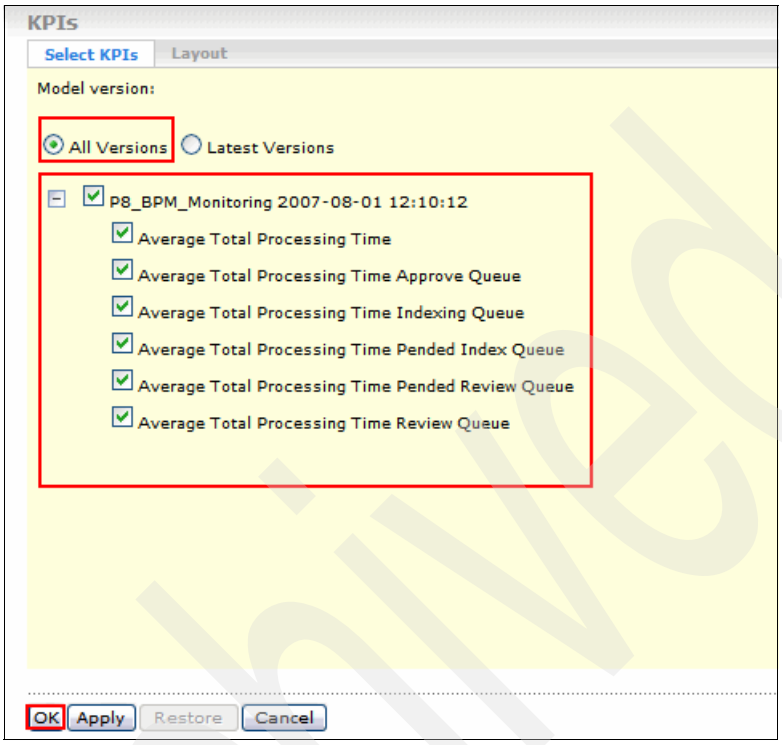


Figure 7-139 KPIs

You see the KPIs dashboard (Figure 7-140).

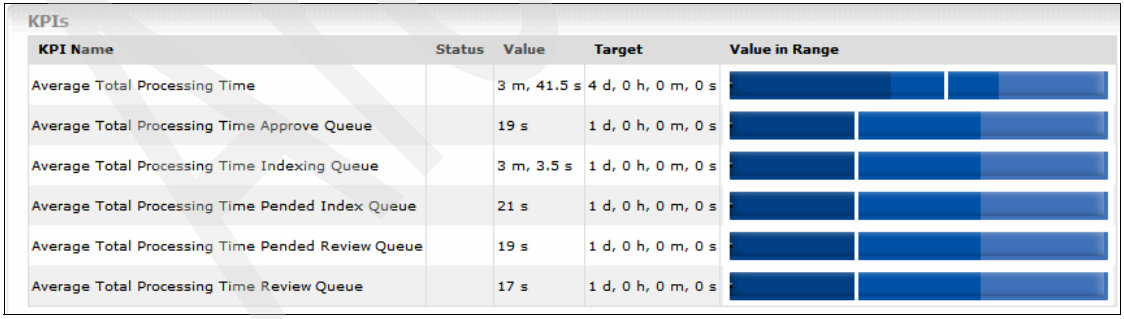


Figure 7-140 KPIs dashboard

Archived

10. Add the Diagrams view to see the diagram of Contract Management Workflow KPIs:

    a. Click **Add to Dashboard**.
    b. On the Add to Dashboard page, select **Diagrams** and click **OK**.
    c. Click **Personalize**.
    d. On the Select a Diagram page (Figure 7-141):

        i. For Model, select **P8 BPM Monitoring <Version>**.

        ii. Under Context, select **Key Performance Indicator context**. Then select **Average Total Processing Time**.

        iii. Click **OK**.

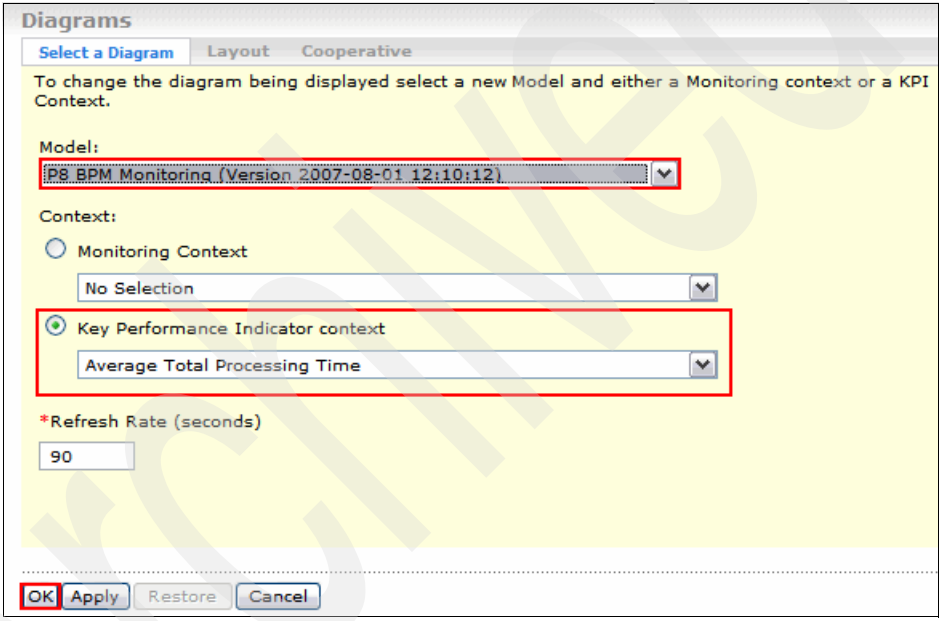


Figure 7-141 Diagrams

Archived

You see the diagram of Average Total Processing Time KPI (Figure 7-142).

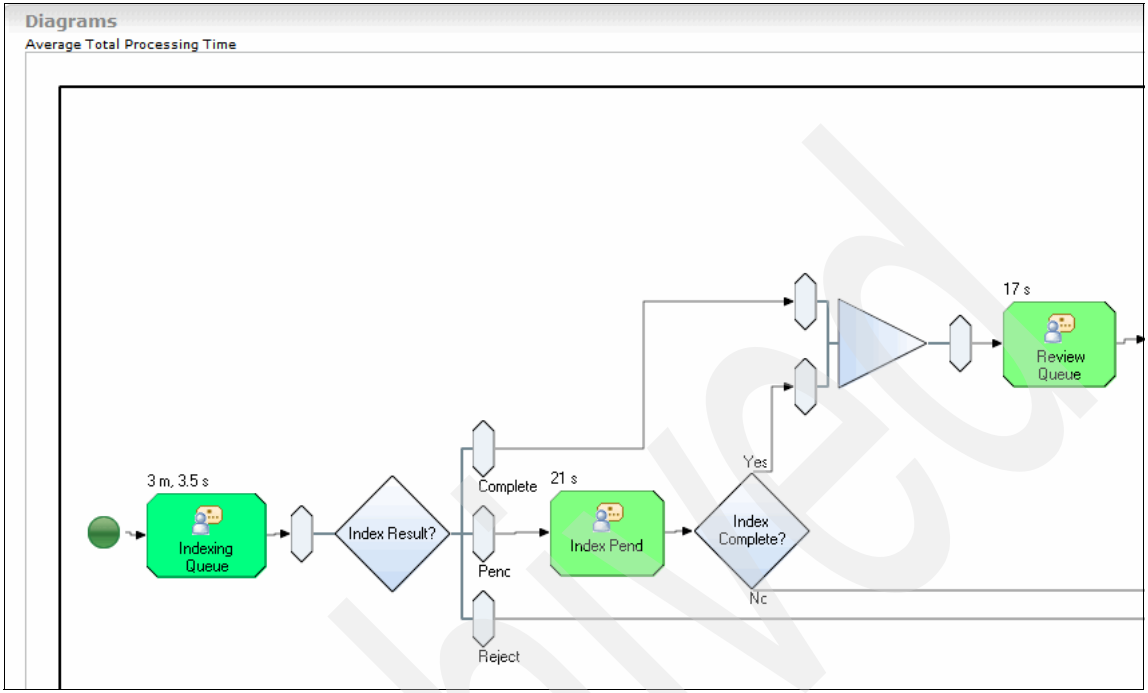


Figure 7-142 Diagrams

11. Add the Alerts view. Click **Add to Dashboard**, select **Alerts**, and click **OK**.

    If there are contracts processed more than four days, you see the alert dashboard as shown in Figure 7-143.

Figure 7-143 Alerts dashboard

7.14 Best practices for monitoring FileNet P8 BPM using IBM WebSphere Business Monitor

There are several best practices for using the FileNet P8 BPM Monitor Model for WebSphere and FileNet P8 BPM Common Base Event Adapter for WebSphere Business Monitor to monitor FileNet P8 BPM using IBM WebSphere Business Monitor.

7.14.1 FileNet P8 BPM Monitor Model for WebSphere

You can customize the FileNet P8 BPM Monitor Model for WebSphere in the following ways:

- Inbound event filter

  Configure the event filter for each inbound event so that the event's Process StartTime is greater than Monitor Model's Deploy DateTime, for example:

  ```
  Activity_Locked/extendedData/processStartTime >
  dateTime('2007-07-31T14:47:27-07:00')
  ```

  In this example, the dateTime represents 31 July 2007 at 2:47 and 27 seconds PM. Subtract 7 from UTC - Pacific Daylight or Mountain Standard. The format of the dateTime is `yyyy-mm-ddThh:mm:ss-zz:zz`.

  This setting specifies that only processes that start after the specified time are considered by the monitor model. It avoids the problem of trying to evaluate processes that were started before the monitor model was deployed. Trying to evaluate such processes causes error messages since the monitor most likely has missed some events that were emitted by the process.

  Tip: You can make this change quickly in the XML view by searching and replacing all instances of the 'MMInstallTime' text with the dateTime() value.

- Add user-defined fields for monitoring, event definitions

  Always add the user-defined field to the P8.BPM.Base event. Before using it in the monitor model, add it to the ExtendedData section as a child element of businessData.

- Add a user-defined field for monitoring, monitor model

  When adding a user-defined field for monitoring, use the exact Inbound Event to set its value. For example, if you have a `CreditApproved` user-defined field in your process and you know its value will be set at the end of the Credit

Approval step, in the monitor model, reference the Inbound Event `BPM Activity End` in its value map, for example `Activity_End/extendedData/businessData/CreditApproved`.

7.14.2 Process engine software

Allow workflows that run on the process engine to terminate when all work is complete. Some customers keep the workflow alive even when all work relating to it is complete in order to maintain records of past activities that corresponding statistical information is kept in the monitor's database tables. If workflows are never terminated, the tables corresponding to processes or activities gradually grow to millions of rows. At that point, the monitor spends more time inserting and updating the database tables. Eventually the Monitor server fails to keep up with the rate of events that the process engine server generates.

Therefore, it is best to terminate the workflow after it is complete and use a different mechanism to maintain records of the completed workflows.

7.14.3 DB2 database for Monitor server

The database administrator must periodically delete the terminated workflows and work items from the Monitor database. If the table is not pruned regularly, the size of the database ultimately slows the performance of the Monitor server.

7.14.4 FileNet P8 BPM Monitoring Dashboard for WebSphere Business Monitor

The FileNet P8 BPM Monitor Model for WebSphere is intended to monitor *work in progress* (that is, active workflows and work items). Therefore, all the views (dimensional, KPI, and so on) must use *TerminationTime* as a dimension and show only data based on TerminationTime= 9999.

7.15 Summary

In this chapter, we explained how to integrate FileNet P8 BPM with WebSphere Business Monitor. We explained the steps to the install FileNet P8 BPM Common Base Event Adapter, create a monitor model, deploy a monitor model, and create a monitor dashboard for Contract Management Workflow.

Archive

Archived

8

WebSphere MQ Workflow

In this chapter, we discuss the architecture of WebSphere MQ Workflow support in WebSphere Business Monitor V6.1. We also explain how to monitor a WebSphere MQ Workflow process with WebSphere Business Monitor V6.1.

Archived

© Copyright IBM Corp. 2008. All rights reserved.

8.1 Introduction to WebSphere MQ

WebSphere MQ Workflow is a powerful business integration tool that is used for quickly building up a workflow management system. The routing features and data control supported by WebSphere MQ Workflow make it possible to focus on modeling and defining the process logic. Whenever changes to the process flow must be done, the applications that are part of the process model do not need to be changed, and we can reuse our software components in other processes. Consequently, we can benefit from minimizing the time required for execution and achieving significant cost savings.

WebSphere MQ Workflow has dedicated components that are responsible for the different workflow management tasks, including the following components:

- WebSphere MQ Workflow Buildtime
- WebSphere MQ Workflow execution server

Normally, *WebSphere MQ Workflow Buildtime* is used to define business process and generate a Flow Definition Language (FDL) file that contains all the information about the business process. With WebSphere MQ Workflow Buildtime, we can create a graphical representation of our processes. We depict our business activities and add the staff that performs them and the programs and network infrastructure that support the people. We also define the flow of control and information between the activities. This modeling information is stored in the relational database of WebSphere MQ Workflow. WebSphere MQ Workflow Buildtime can also export FDL files. However, because the FDL description is public, FDL files can be generated in other ways, such as by WebSphere Business Integration Workbench, which is another tool for modeling and exporting FDL files.

The most important component for interoperability is the *WebSphere MQ Workflow execution server*, which is responsible for starting and navigating business processes. We can make the information about business processes, the infrastructure, and participating people (staff) known to the execution server by populating the runtime database with the corresponding information. We do this by importing an FDL file and using a tool that translates the FDL definitions into the entities of the runtime database. For every instance of the process, the execution server navigates through the process and moves the work to the right person in the right sequence. WebSphere MQ Workflow starts the programs, keeps process execution history, and provides recovery and restart procedures.

During the process execution, activities that need to be performed are displayed in worklists of the WebSphere MQ Workflow client or the browser-based Web client of assigned staff members. When a staff member selects, for example, a program activity, the program is started with the necessary information. Users'

worklists contain continuously updated overviews of their pending activities. As we implement workflow management, people can use worklists as their primary user interface to other applications. They can then access applications and data on different platforms and user interfaces.

WebSphere MQ Workflow also provides APIs to programmers who can modify applications to be attached to process models or develop new applications that can be fully automated or with little human guidance.

8.2 Overview of WebSphere MQ Workflow support in WebSphere Business Monitor V6.1

In WebSphere Business Monitor V6.1, the capability to monitor WebSphere MQ Workflow processes is enabled. In this section, we present an architectural overview of WebSphere Business Monitor V6.1 support to monitor WebSphere MQ Workflow processes. We discuss the following topics:

- The architecture from modeling time and runtime points of view respectively
- Possible installation topologies
- Software and hardware prerequisites

8.2.1 Architecture of WebSphere MQ Workflow support

In this section, we describe the modeling time development environment and the runtime environment of WebSphere MQ Workflow as we use it in this chapter.

Modeling time

A new tool named *FDL to monitor model utility for WebSphere MQ Workflow* is provided in the WebSphere Business Monitor V6.1 development toolkit package. This tool is also referred to by using the shortened name *FDL to monitor model utility.*

With the FDL to monitor model utility, we can import an FDL file and generate the monitor model based on the imported FDL file. During the monitor model generation, the monitoring templates are provided by FDL to the monitor model utility for the quick generation of general measures.

The FDL to monitor model utility runs as a Rational Application Developer or a WebSphere Integration Developer plug-in. Therefore, ensure that Rational Application Developer or WebSphere Integration Developer with the WebSphere Business Monitor development toolkit is installed before you install the FDL to monitor model utility.

Figure 8-1 on page 484 illustrates the steps to develop a monitor model for a WebSphere MQ Workflow process, which are explained as follows:

1. Implement the WebSphere MQ Workflow process and export the .fdl file.
2. Enable event filtering in the .fdl file for performance reasons. If the .fdl file is exported from WebSphere Business Integration Workbench, event filtering can be set before export. If the .fdl file is exported from WebSphere MQ Workflow Buildtime, we can use a text editor to set event filtering.
3. Import the .fdl file by using the FDL to monitor model utility.
4. Generate the monitor model and select the necessary monitoring events and templates.
5. Augment the monitor model with more measures and export the deployable enterprise archive (EAR) file.

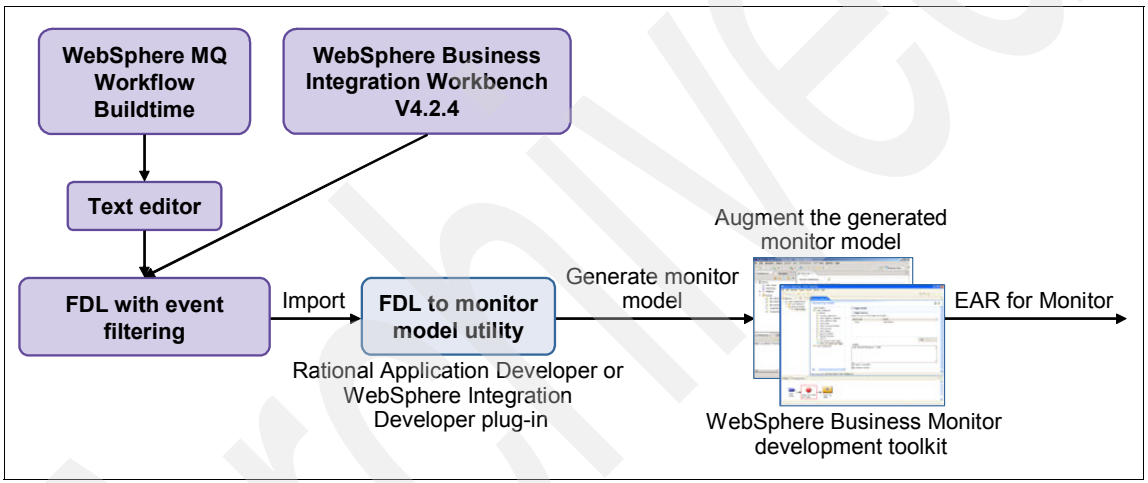


Figure 8-1 Modeling time architecture

Run time

With the WebSphere MQ Workflow ServicePack 6 enhancement, WebSphere MQ Workflow run time can write container data in an audit trail that is sent to a WebSphere MQ queue. Later the monitor application can get the necessary container data while processing the WebSphere MQ Workflow process Common Base Events and use them to set values for measures. You can find WebSphere MQ Workflow ServicePack 6 at the following Web address:

http://www-1.ibm.com/support/docview.wss?rs=795&context=SSVLA5&dc=D400&uid=swg24010935&loc=en_US&cs=UTF-8&lang=en

Archived

The WebSphere MQ Workflow Event Converter, shipped in WebSphere MQ Workflow SupportPac WA61, is a standard J2EE application that is deployed on a WebSphere Application Server V6.1. The WebSphere MQ Workflow Event Converter is responsible for transforming the audit trail from the WebSphere MQ Workflow audit trail queue into Common Base Events and publishing them to a Common Event Infrastructure (CEI) server. Later a monitor application can receive the Common Base Events and process them for monitoring. You can find WebSphere MQ Workflow SupportPac WA61 at the following Web address:

http://www-1.ibm.com/support/docview.wss?rs=171&uid=swg24017759&loc=en_US&cs=utf-8&lang=en

Figure 8-2 on page 486 illustrates the general process of WebSphere Business Monitor monitoring WebSphere MQ Workflow, which is explained as follows:

1. Deploy the FDL into the WebSphere MQ Workflow run time.
2. Whenever a process runs in the WebSphere MQ Workflow run time, the workflow engine writes audit trail data to a WebSphere MQ queue.
3. The WebSphere MQ Workflow Event Converter reads the data from the audit trail queue and transforms it into Common Base Events, which are sent out by using the CEI.
4. WebSphere Business Monitor consumes and processes the Common Base Events.
5. The collected data is shown in the WebSphere Business Monitor dashboards.

Archived

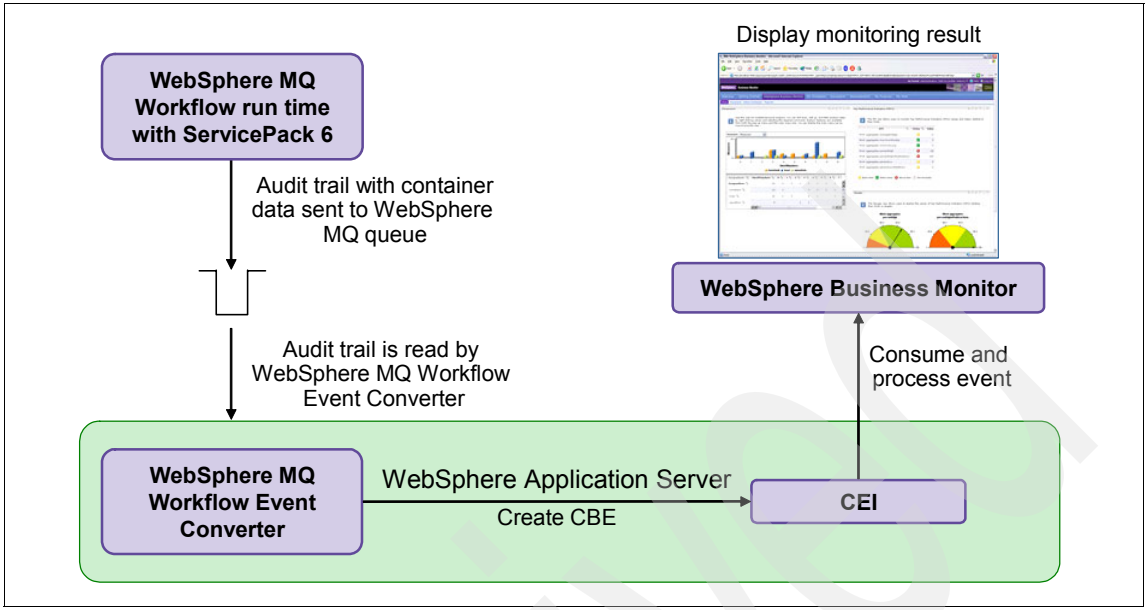


Figure 8-2 Runtime architecture

8.2.2 Installation topologies

WebSphere MQ Workflow, WebSphere MQ Workflow Event Converter, and WebSphere Application Server can be installed on different computers. We provide a brief introduction to the various options.

CEI server: WebSphere MQ Workflow Event Converter is installed on WebSphere Application Server. It emits Common Base Events to the CEI within the same WebSphere Application Server. Therefore, the WebSphere Application Server is also used as the CEI server.

Since the WebSphere Business Monitor V6.1 server is installed upon WebSphere Application Server, we can install WebSphere MQ Workflow Event Converter on the V6.1 server. As a result, the CEI server and the Monitor server reside in the same WebSphere Application Server and integrate naturally. However, we can also install a stand-alone WebSphere Business Monitor server and integrate the CEI server with the WebSphere Business Monitor server.

Single-box setup

In a single-box setup, WebSphere MQ Workflow, WebSphere MQ Workflow Event Converter, and WebSphere Application Server are installed on one machine and use the same queue manager (Figure 8-3).

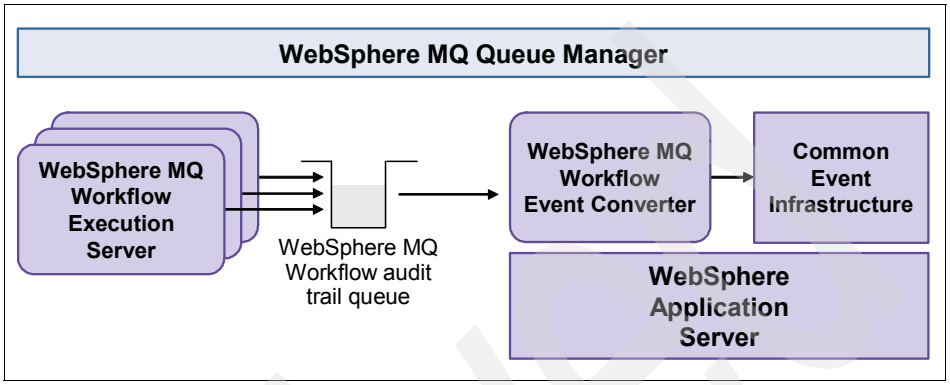


Figure 8-3 Single-box setup

Multiple-box setup with different queue managers

In a multiple-box setup with different queue managers (Figure 8-4 on page 487), WebSphere MQ Workflow is installed on the machine different from the one for WebSphere Application Server and the WebSphere MQ Workflow Event Converter. Separate queue managers are used. The WebSphere MQ Workflow audit trail queue is located on the same machine as WebSphere Application Server. We must connect WebSphere MQ Workflow to the WebSphere MQ Workflow audit trail queue by use of MQ clustering.

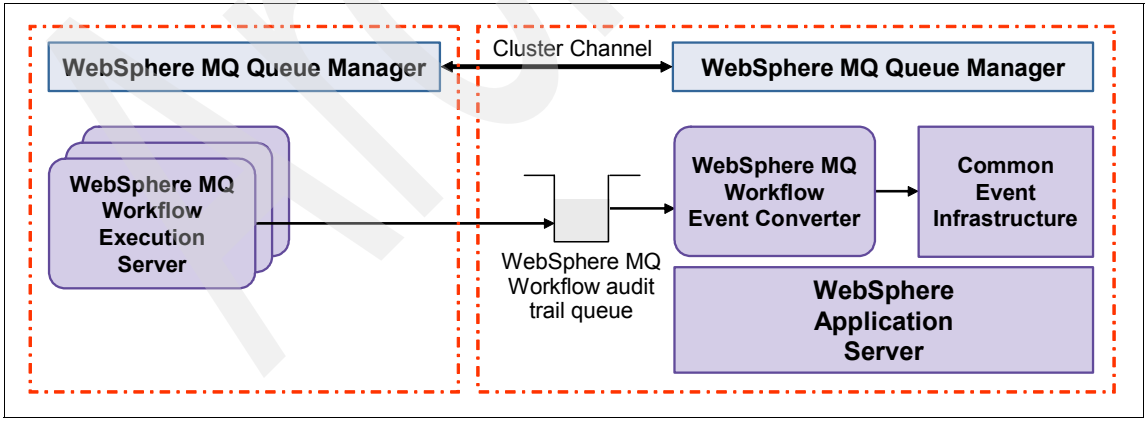


Figure 8-4 Multiple-box setup with different queue managers

Multiple-box setup with a shared queue manager

In a multiple-box setup with a shared queue manager (Figure 8-5), WebSphere MQ Workflow is installed on a machine different from the one for WebSphere Application Server and the WebSphere MQ Workflow Event Converter. The queue manager with the WebSphere MQ Workflow audit trail queue is located on the same machine as WebSphere MQ Workflow. The WebSphere MQ Workflow Event Converter accesses the WebSphere MQ Workflow audit trail queue through the WebSphere MQ Client channel.

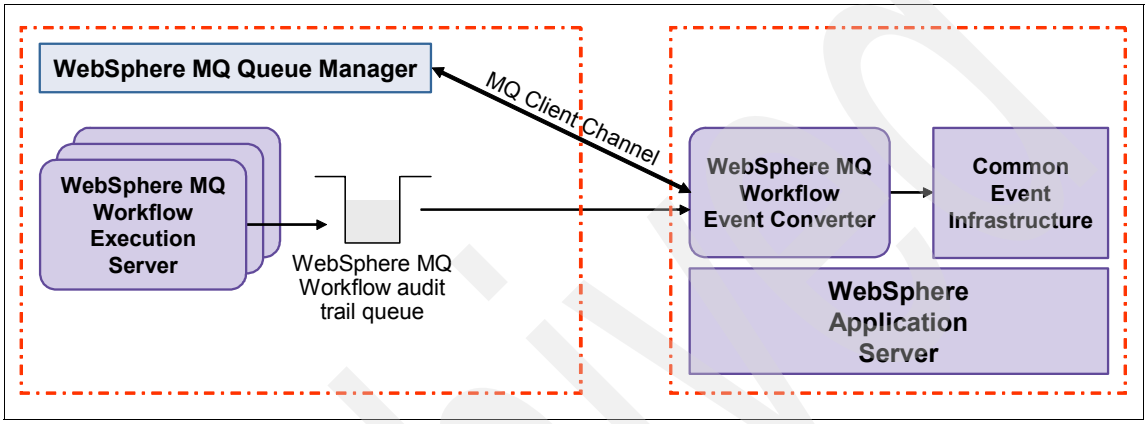


Figure 8-5 Multiple-box setup with a shared queue manager

8.2.3 Prerequisites for WebSphere MQ Workflow support

WebSphere MQ Workflow support in WebSphere Business Monitor V6.1 entails the following software and hardware prerequisites.

Software prerequisites

You must ensure that the following software is installed:

- WebSphere Business Monitor V6.1 and its prerequisites
- WebSphere MQ Workflow V3.6 and its prerequisites
- WebSphere MQ V6.0 or higher
- WebSphere MQ Workflow ServicePack V6
- WebSphere MQ Workflow SupportPac WA61
- WebSphere Application Server V6.1

Hardware prerequisites

It is possible to install DB2, WebSphere MQ, WebSphere MQ Workflow, WebSphere Business Monitor, and WebSphere MQ Workflow Event Converter

on one machine. This type of installation requires the maximum hardware prerequisites. The following parameters are suggested:

- 2 GHz CPU (minimum)
- 3 GB RAM (minimum)
- 20 GB of disk space (minimum)

8.3 Setting up the runtime environment

In this section, we discuss the installation and configuration to enable the integration of WebSphere MQ Workflow with WebSphere Business Monitor. We introduce the topology used in the sample. We explain how to apply WebSphere MQ Workflow ServicePack V6 and WebSphere MQ Workflow Event Converter.

Before starting the sample, we assume that DB2, WebSphere MQ, WebSphere MQ Workflow, and WebSphere Business Monitor are installed correctly.

8.3.1 Runtime topology for the sample

In this sample, we use a two box setup with a shared queue manager and install WebSphere MQ Workflow Event Converter on WebSphere Business Monitor V6.1.1 server. Figure 8-6 illustrates this topology.

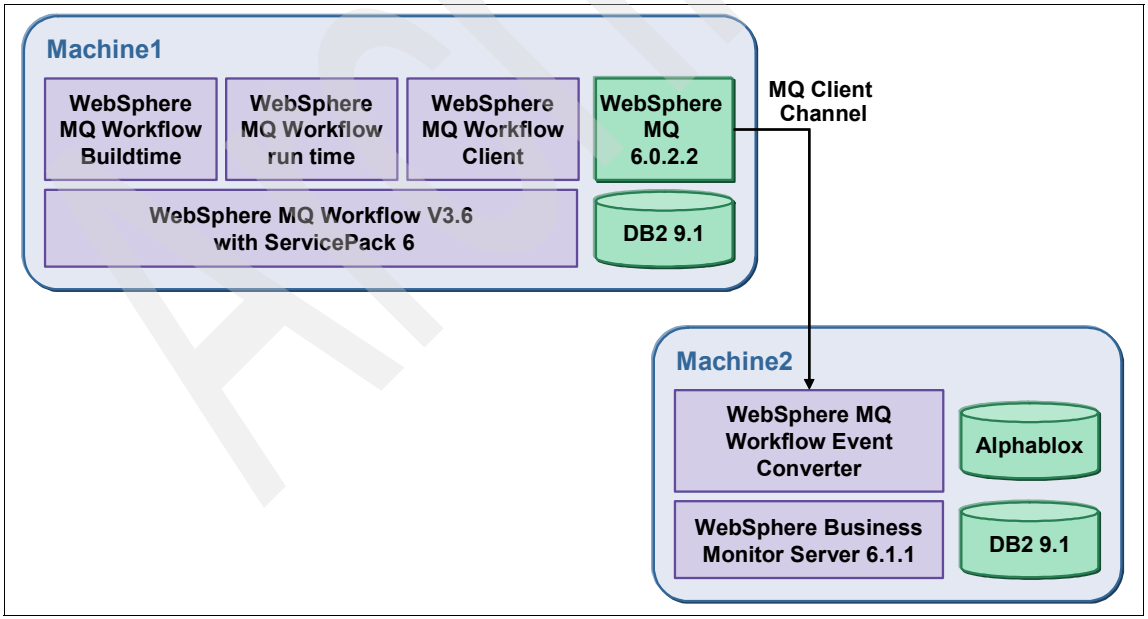


Figure 8-6 Topology used in the sample

Archived

The following software versions are used in this sample:

- DB2Universal Database V9.1
- WebSphere MQ V6.0.2.2
- WebSphere MQ Workflow V3.6
- WebSphere MQ Workflow ServicePack 6
- WebSphere MQ Workflow SupportPac WA61
- WebSphere Business Monitor V6.1.1
- WebSphere Integration Developer V6.1

8.3.2 Applying WebSphere MQ Workflow ServicePack 6

In this sample, we install the WebSphere MQ Workflow V3.6 full package on one machine. Therefore, we must download WebSphere MQ Workflow for Windows 3.6.0 ServicePack 6 (Full Pkg) and apply it.

To apply WebSphere MQ Workflow ServicePack 6:

1. Stop all running WebSphere MQ Workflow processes and services. In this sample, we check the Windows services application to ensure that no WebSphere MQ Workflow service is started.
2. Double-click **wf360w6a.exe**.
3. In the IBM WebSphere MQ Workflow Version 3.6 Service Pack 6 window, click **Next**.
4. In the Start Copying Files window (Figure 8-7), click **Next**.

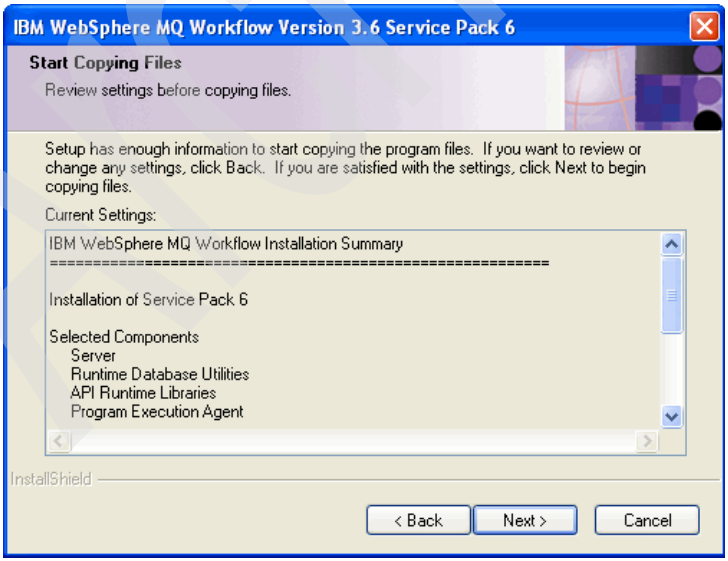


Figure 8-7 WebSphere MQ Workflow ServicePack 6 summary page

5. In the completion window (Figure 8-8), click **Finish** to complete the installation of WebSphere MQ Workflow ServicePack 6.

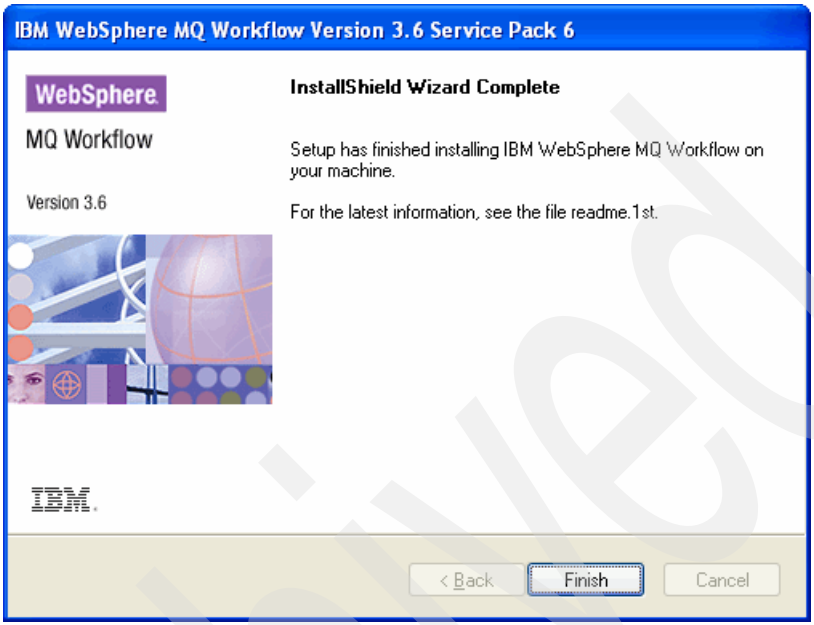


Figure 8-8 WebSphere MQ Workflow ServicePack 6 completion page

Archived

6. In this sample, we use FMC as WebSphere MQ Workflow configuration ID. The MQ Workflow server, buildtime, and runtime client are configured and used as shown in Figure 8-9. To start the configuration utility, click **Start** → **All Programs** → **IBM** → **IBM WebSphere MQ Workflow** → **IBM WebSphere MQ Workflow Configuration Utility**. Accept all the defaults during the configuration.

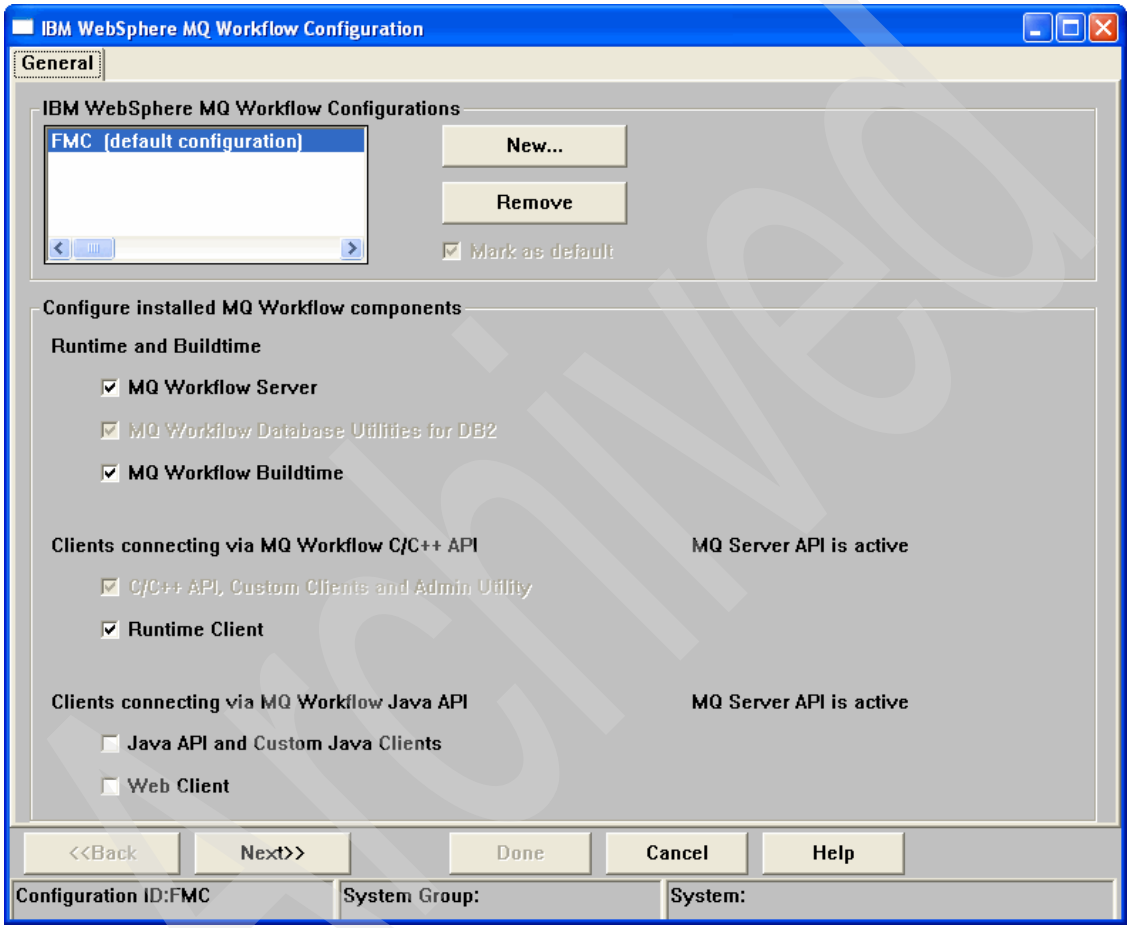


Figure 8-9 WebSphere MQ Workflow configuration

7. When the FMC configuration is ready, open the Command Prompt by running **cmd.exe** so that we can set certain profile variables.
8. Set the EnableContainerDataInAuditTrail variable so that you can write input or output container data in events (Figure 8-10 on page 493):

```
fmczchk -c inst:m,EnableContainerDataInAuditTrail,TRUE -yFMC
```

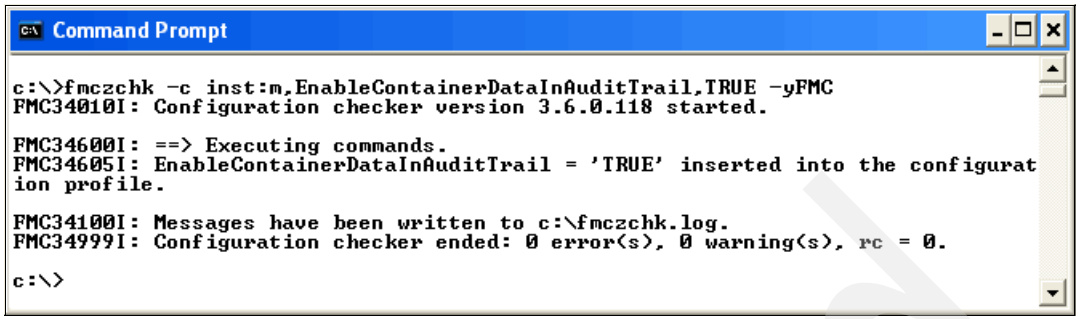


Figure 8-10 Setting the EnableContainerDataInAuditTrail variable

9. Set the RTGlobalClock variable so that you can guarantee the uniqueness of the time stamps in the workflow events (Figure 8-11):

```
fmczchk -c inst:m,RTGlobalClock,TRUE -yFMC
```

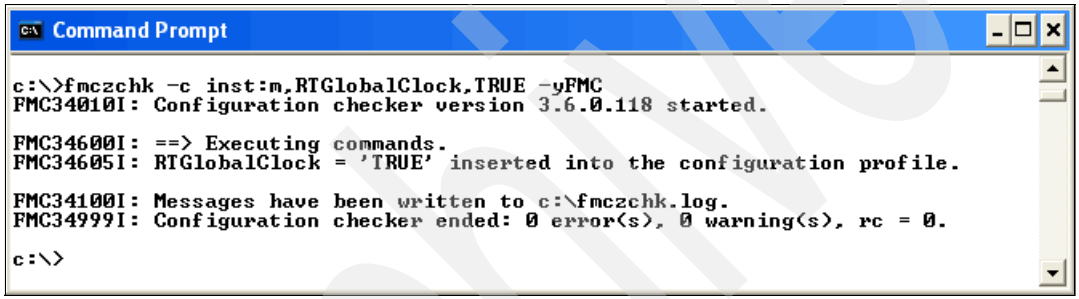


Figure 8-11 Setting the RTGlobalClock variable

10. Set the EnableProcessActivityCorrelation variable so that you can correlate the subprocesses with their calling activities (Figure 8-12):

```
fmczchk -c inst:m,EnableProcessActivityCorrelation,TRUE -yFMC
```

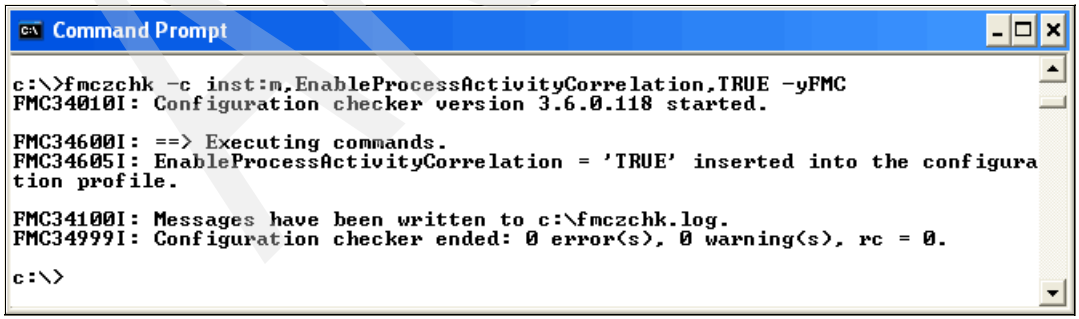


Figure 8-12 Setting the EnableProcessActivityCorrelation variable

Archived

8.3.3 Applying WebSphere MQ Workflow SupportPac WA61

In this sample, we deploy the WebSphere MQ Workflow Event Converter on the WebSphere Business Monitor server with necessary configurations. The Event Converter is in the WebSphere MQ Workflow SupportPac.

Before we install WebSphere MQ Workflow Event Converter, we must create a new queue named FMC.FMCGRP.AUDITQ under the queue manager FMCQM on the WebSphere MQ machine. Later, when we configure the .fdl file, we must enter the same queue name and queue manager name so that the WebSphere MQ Workflow can send the audit trail to this queue and the Event Converter can fetch the audit trail from the same queue. Create the queue by using WebSphere MQ Explorer. In the New Local Queue window (Figure 8-13), accept all default settings and click **Finish**.

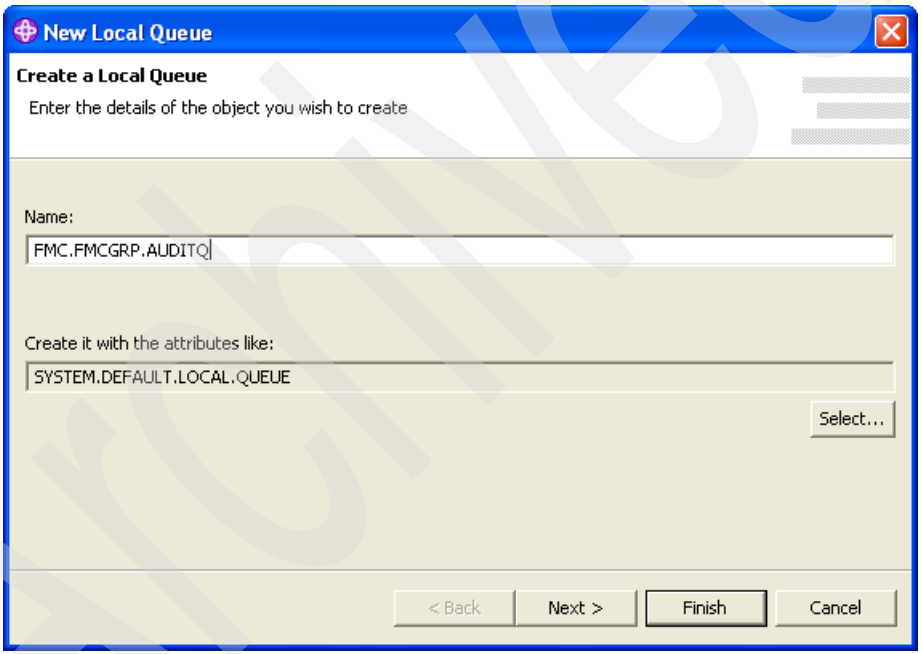


Figure 8-13 Creating a new queue

Archive

To install WebSphere MQ Workflow Event Converter:

1. Switch to the WebSphere Business Monitor server machine and ensure that the Monitor server is started.
2. Modify the *hosts* file in C:\WINDOWS\system32\drivers\etc\ to add the WebSphere MQ machine IP address (Figure 8-14). By doing this, if the WebSphere MQ machine's IP address changes, we must only modify the hosts file instead of reconfiguring the Event Converter parameters. In this sample, we type `RemoteMQ` for the host name.

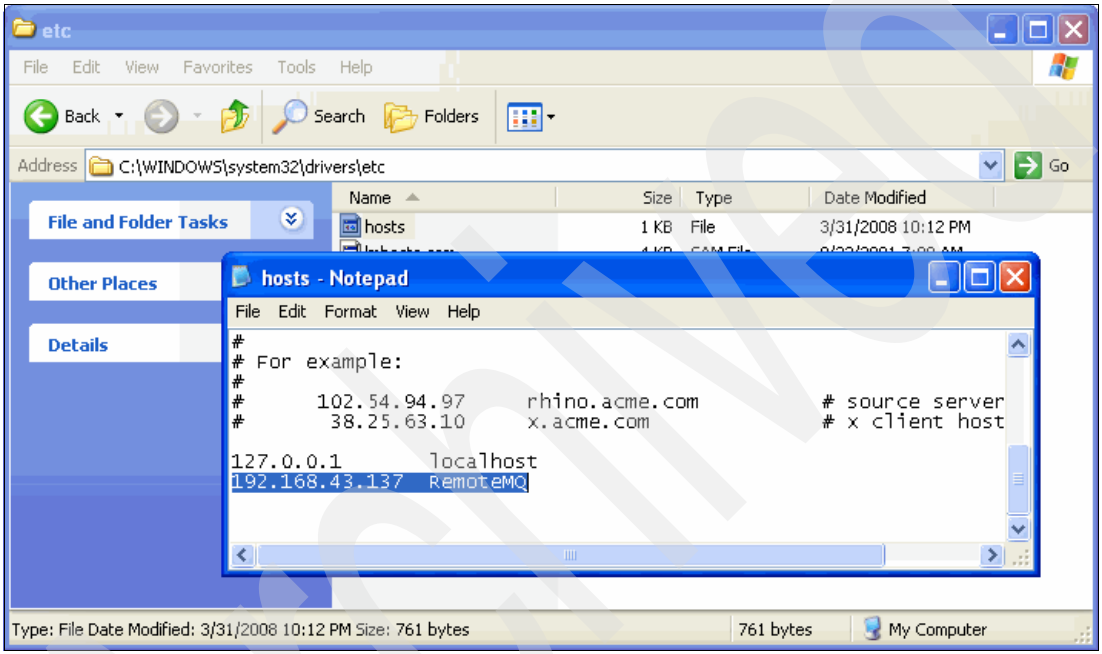


Figure 8-14 Setting the remote WebSphere MQ machine IP and host name

3. Extract the files from the SupportPac WA61 compressed file.

4. Open a Command Prompt window and navigate to the install directory in the directory where you extracted the SupportPac file (Figure 8-15).

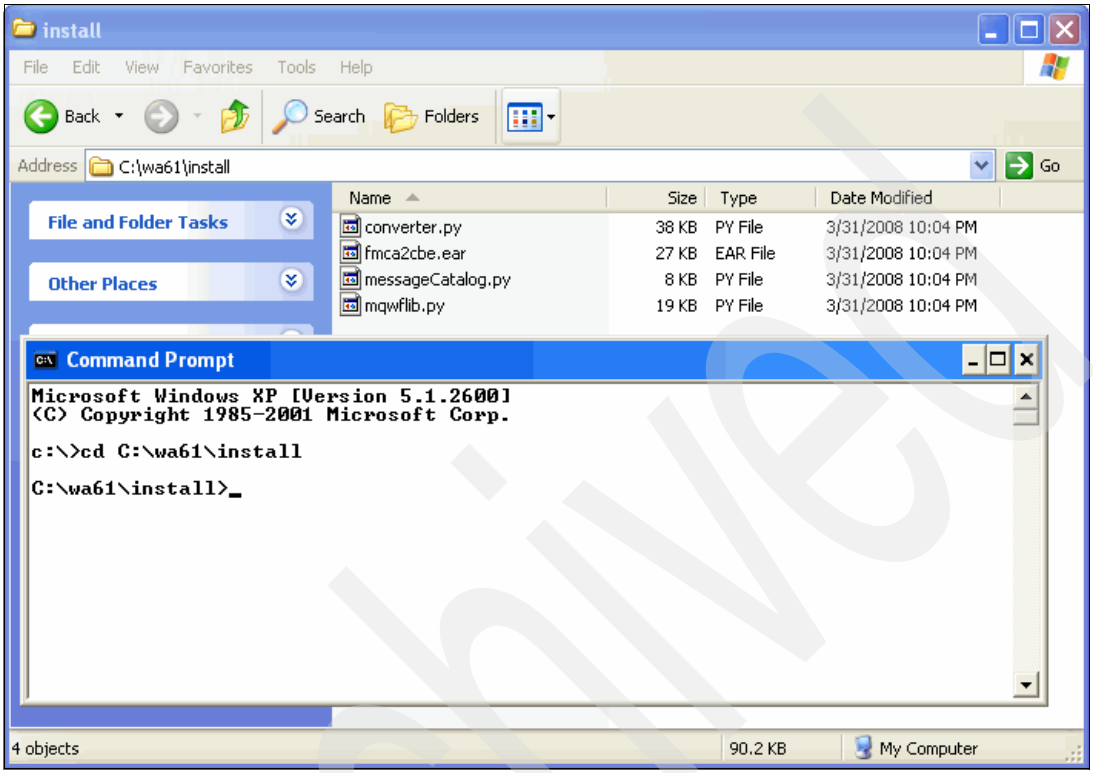


Figure 8-15 Navigating to the install directory

Archive

5. Start the script to deploy the Event Converter (Figure 8-16) by entering the following command from the command line:

```
%WAS_HOME%/bin/wsadmin -f converter.py
```

Figure 8-16 Starting to install Event Converter

Archived

6. Accept the default value for MQ Workflow system group name, queue manager name, audit queue name, queue connection factory, JMS provider, display name, and JNDI name (Figure 8-17). Especially for MQ Workflow audit queue name, the value must be consistent with the one that we created on page 494.

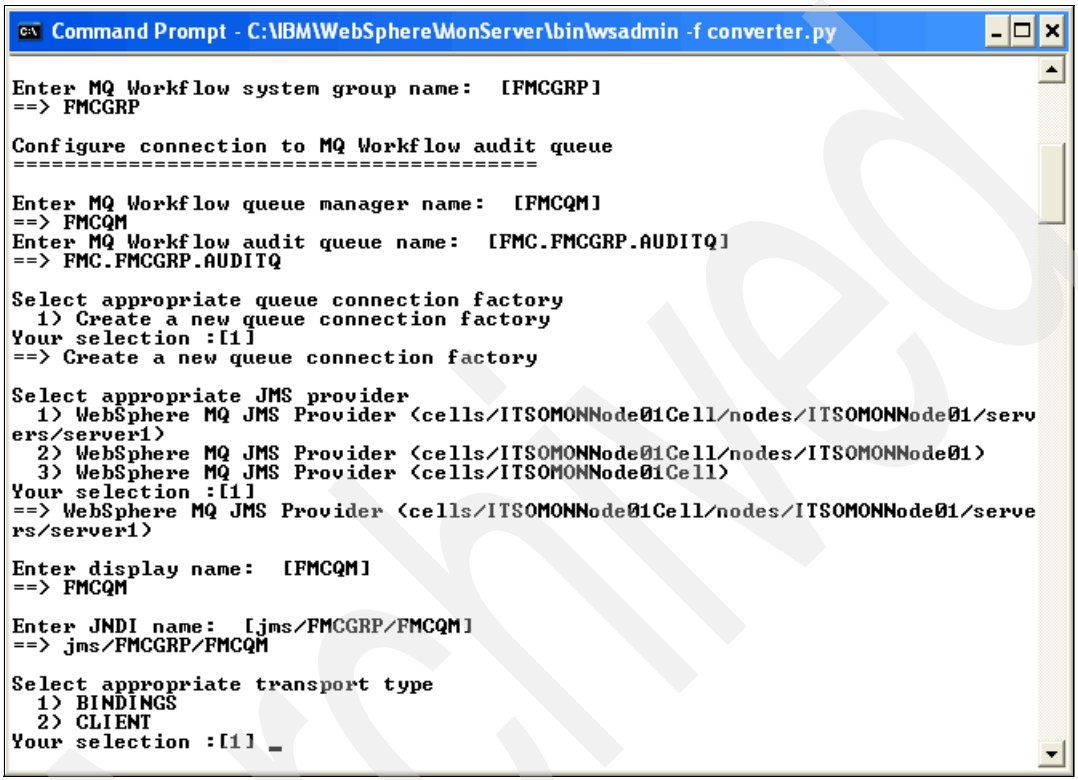


Figure 8-17 Setting the MQ Workflow system group and queue connection factory

Archived

7. Since the WebSphere MQ machine is a remote machine, for transport type, select `CLIENT`. Check the WebSphere MQ configuration for the values for the host name, port number, and channel name of the queue manager to which the WebSphere MQ Workflow belongs (Figure 8-18). Enter them accordingly in the Command Prompt window. In our sample, we enter `RemoteMQ` for hostname and `5010` for port name. We accepted the default for channel name.

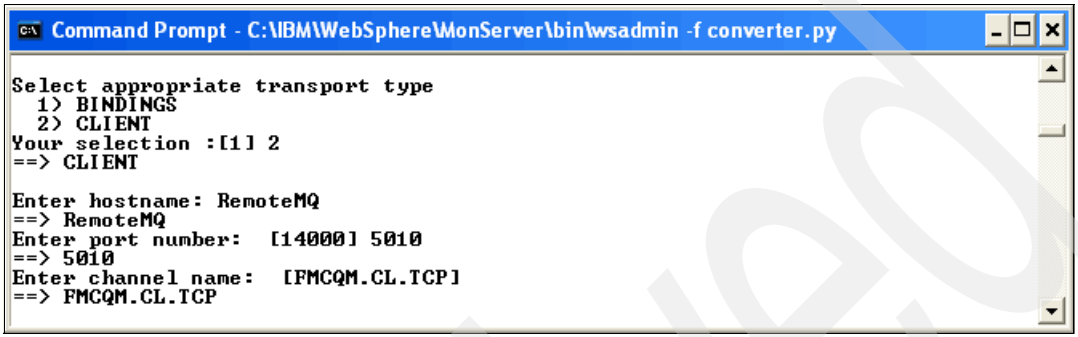


Figure 8-18 Selecting the transport type and setting the corresponding parameters

8. Review the summary of the configurations and type `1` to save the changes (Figure 8-19).

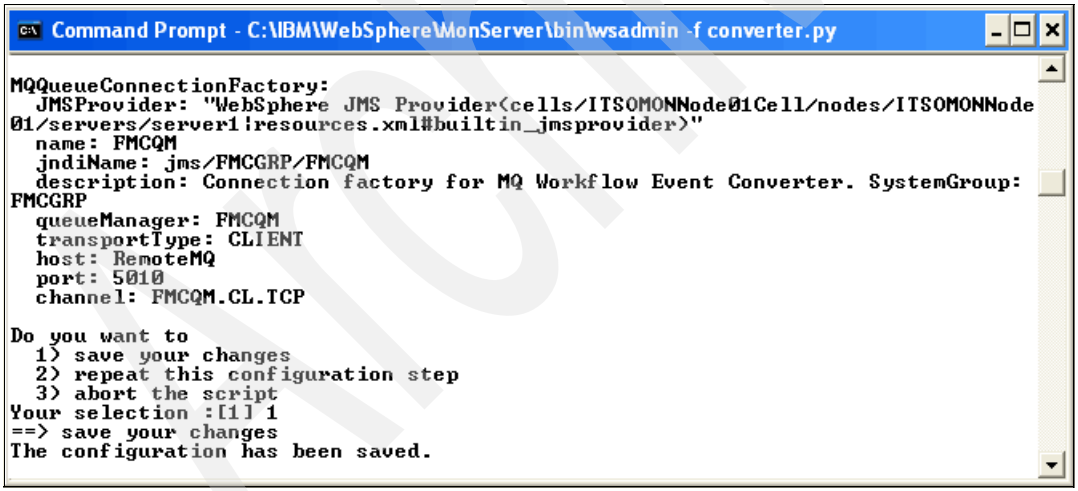


Figure 8-19 Saving the queue connection factory changes

Archived

9. Type 1 to create a new queue definition (Figure 8-20).

Figure 8-20 Creating a new queue definition

10. Accept the default value for a queue's display name and JNDI name (Figure 8-21).

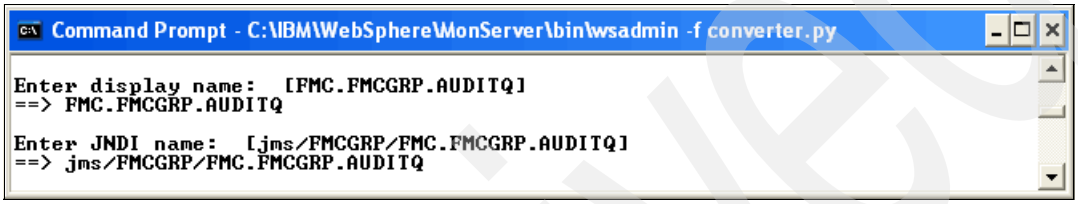


Figure 8-21 Accepting the default values for display name and JNDI name

11. Review the summary of the configurations and type 1 to save the changes (Figure 8-22).

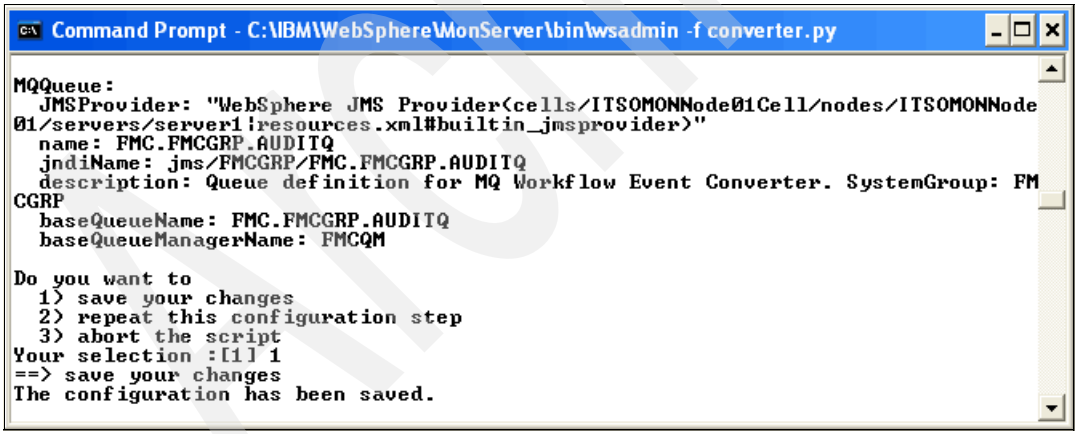


Figure 8-22 Saving the queue definition changes

Archived

12. Accept the default value for listener port and save the changes (Figure 8-23).

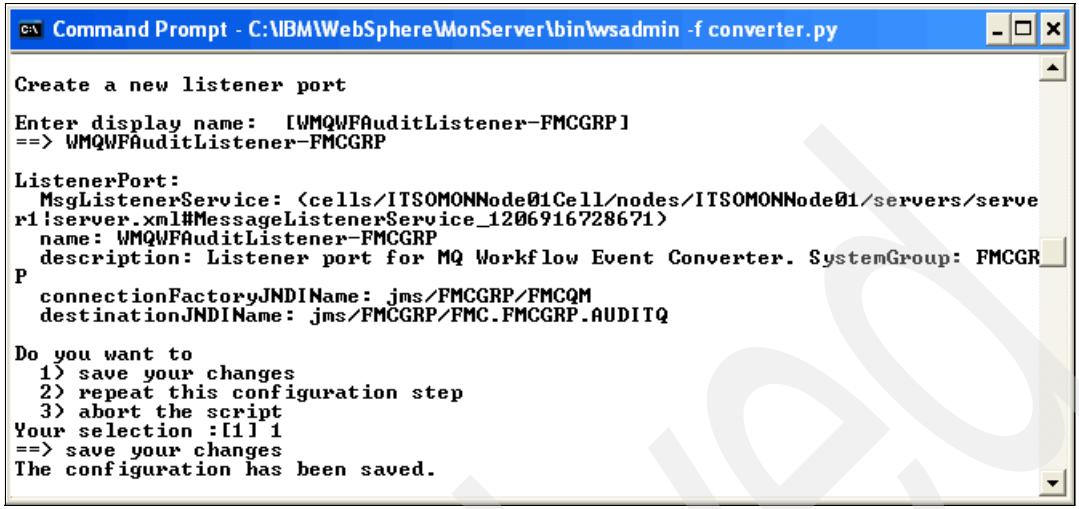


Figure 8-23 Saving the changes for listener port

13. To configure the connection to the CEI, specify the user name and the password that are needed to authenticate the user for the CEI bus. We do this because, in this sample, the security is not enabled for WebSphere Business Monitor server. Then type 1 for the CEI emitter factory profile (Figure 8-24).

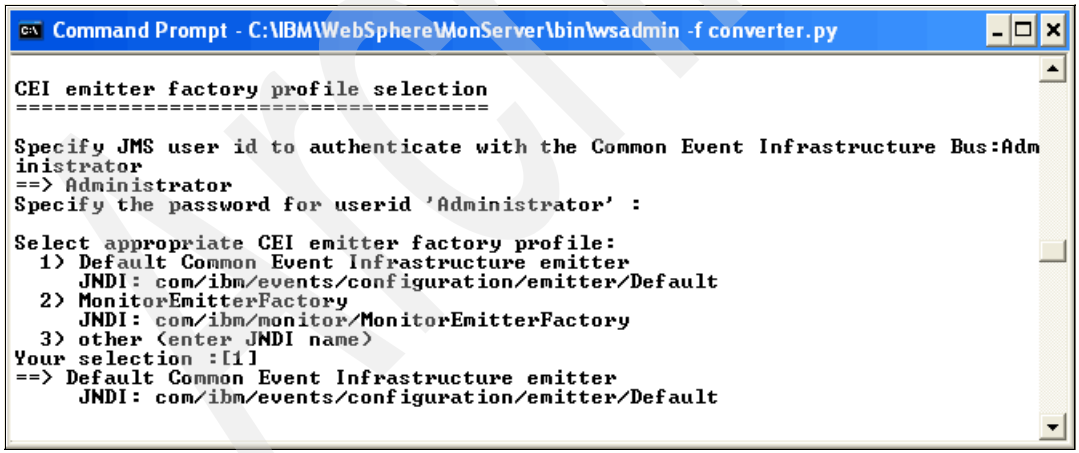


Figure 8-24 Configuring the CEI parameters

14. The installation of the Event Converter starts and the summary are displayed after a while (Figure 8-25). Type y to save the changes.

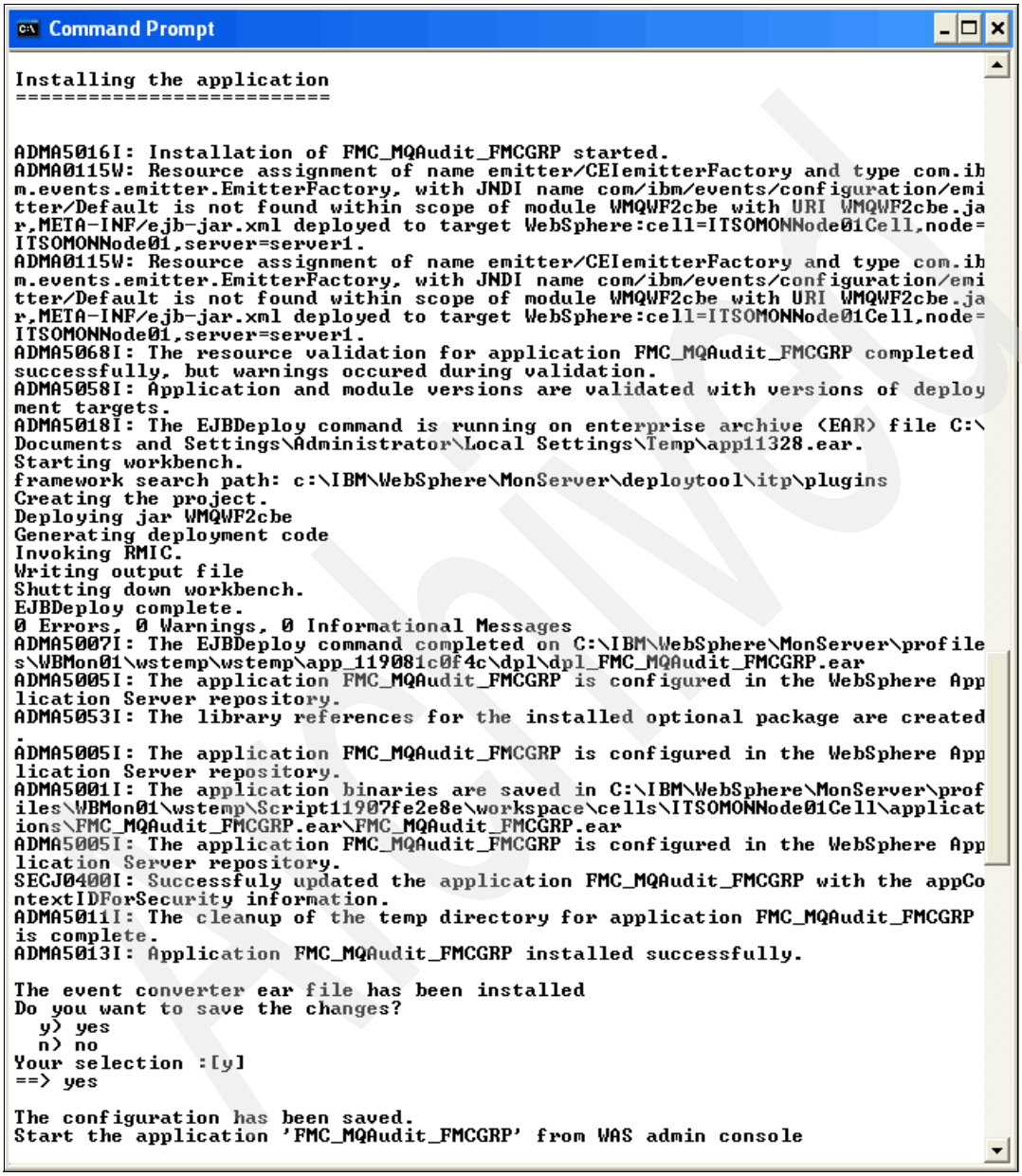


Figure 8-25 Installation of the Event Converter

15. Restart the WebSphere Business Monitor server.

16. Open the WebSphere Business Monitor server administrative console.

17. In the Integrated Solutions Console, in the left pane, expand **Applications** and click **Enterprise Applications**. Verify that the Event Converter application FMC_MQAudit_FMCGRP starts correctly (Figure 8-26). Of course, the remote WebSphere MQ machine starts at this time so that Event Converter can connect and start correctly.

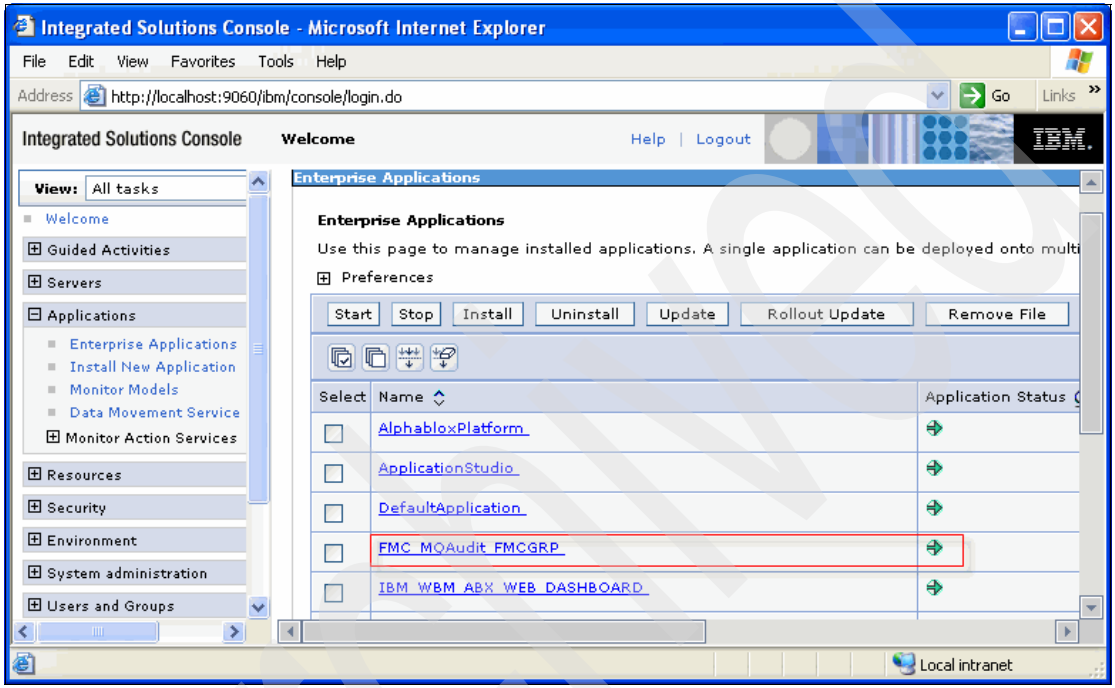


Figure 8-26 Verifying that the Event Converter is installed correctly

8.4 Implementing the scenario by using WebSphere MQ Workflow

In this section, we implement the ClipsAndTacks scenario by using WebSphere MQ Workflow. While we do not explain the detailed steps, we include several windows that illustrate the final implemented process. The .fdl file is supplied in the additional materials for this book in the \WMQWF directory and can be imported directly. For more information about accessing this file, see Appendix B, “Additional material” on page 801.

8.4.1 Implementation of the WebSphere MQ Workflow process

Figure 8-27 illustrates how the WebSphere MQ Workflow process is implemented.

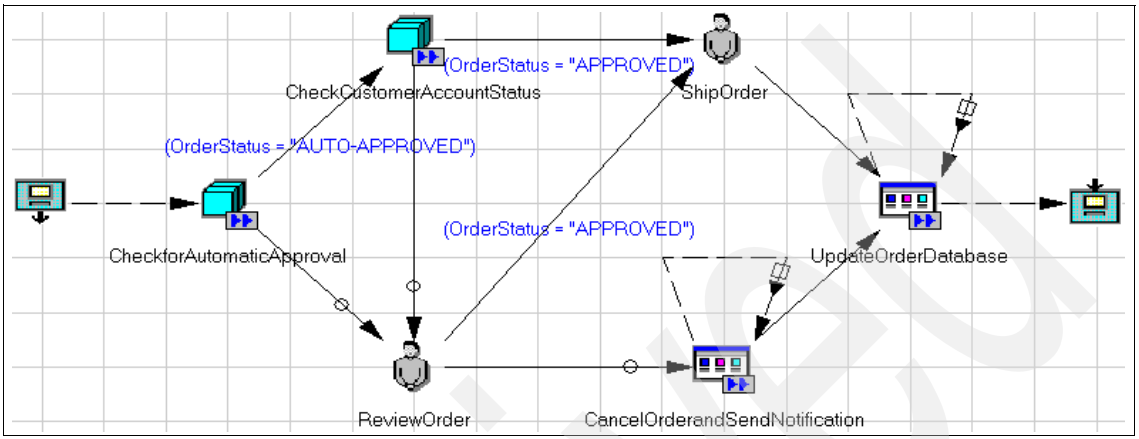


Figure 8-27 WebSphere MQ Workflow process implementation

Archived

The CheckforAutomaticApproval activity is implemented through the Block activity, which is implemented as shown in Figure 8-28.

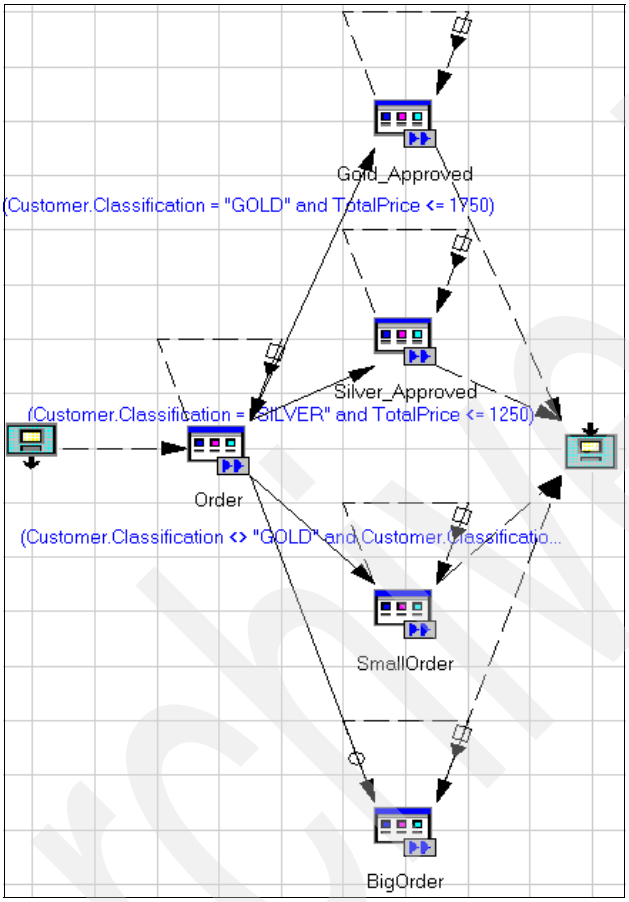


Figure 8-28 CheckforAutomaticApproval Block activity implementation

Archived

The CheckCustomerAccountStatus activity is implemented through a Block activity, which is implemented as shown in Figure 8-29.

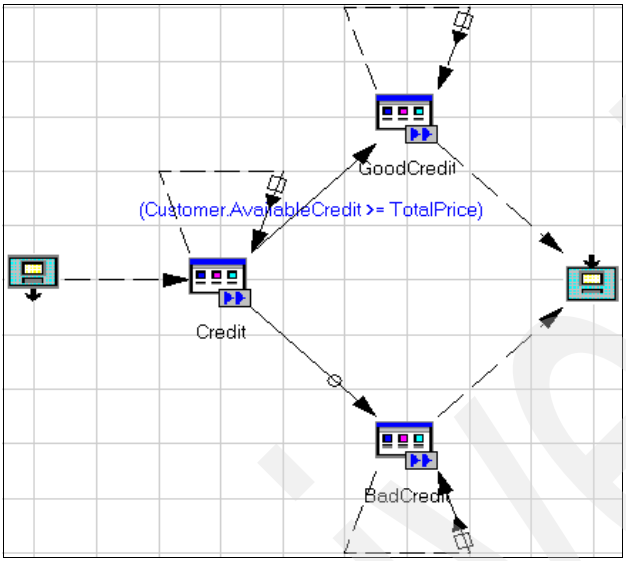


Figure 8-29 CheckCustomerAccountStatus Block activity implementation

8.4.2 Use cases of the process

Table 8-1 shows the various execution paths with different data to help you understand the process execution behavior when we exercise the process later.

Table 8-1 Use cases of the WebSphere MQ Workflow process

Condition (And)		Next activity	Condition	Next activity	Condition	Next activity
Customer classification	**Total price**					
GOLD	<=1750	Check Customer Account Status	Customer.AvailableCredit >=TotalPrice	Ship Order		
SILVER	<=1250					
Any	<=750		Else	Review Order	OrderStatus = "APPROVED"	ShipOrder
						CancelOrder

Archived

Condition (And)		Next activity	Condition	Next activity	Condition	Next activity
Customer classification	Total price					
Else		Review Order	OrderStatus = "APPROVED"	Ship Order		
			Else	Cancel Order		

8.4.3 Setting the queue and queue manager name

We must define the queue name and the queue manager name for our WebSphere MQ Workflow system group which is the target for the WebSphere MQ Workflow audit messages:

1. Open the **OrderHandling.fdl** file in a text editor. This file contains the implemented WebSphere MQ Workflow process. It is provided in the additional material for this book in the \WMQWF directory. For information about how to download the materials, see Appendix B, “Additional material” on page 801.
2. Find the UPDATE GROUP ‘FMCGRP’ part in the .fdl file.

   Notice the queue and queue manager definition. Example 8-1 shows the code required in the .fdl file. The queue name is the same as we created in 8.3.3, “Applying WebSphere MQ Workflow SupportPac WA61” on page 494.

Example 8-1 Setting the queue and queue manager names for the system group

```
UPDATE GROUP 'FMCGRP'
  DESCRIPTION "Default system group"
  AUDIT_QUEUE_MANAGER_NAME "FMCQM"
  AUDIT_QUEUE_NAME "FMC.FMCGRP.AUDITQ"
  RELATED_DOMAIN 'DOMAIN'
  OPERATION
    SYSTEM_QUALIFIER "FMC"
END 'FMCGRP'
```

Archived

8.4.4 Enabling the audit to WebSphere MQ

To enable the audit to WebSphere MQ and, for performance reasons, to enable filtering:

1. Open the **OrderHandling.fdl** file in any text editor.
2. Set the audit to WebSphere MQ and filtering at the process level. Find the **PROCESS 'OrderHandling'** part in the .fdl file.

   Notice the filter definition. Example 8-2 shows the code required in the .fdl file.

Example 8-2 Enabling audit to WebSphere MQ

```
PROCESS 'OrderHandling' ( 'Order', 'Order' )
  ...
  DO NOT PROMPT_AT_PROCESS_START
  FILTER AUDIT_TO_MQ
  AUDIT_FILTER_MQ "21017 21018 21000 21020 21025 21026 21015 21016
21006 21007 21011"
  WINDOW VIEW_REFERENCE_POINT XPOS 0 YPOS 293
```

Table 8-2 explains when the events are emitted.

Table 8-2 Enabled event codes and descriptions

Event code	Event description
21017	Process created
21018	Process created and started
21000	Process started
21020	Process deleted
21025	Process ended normally and deleted
21026	Process terminated and deleted
21015	Block started
21016	Block ended
21006	Activity ready
21007	Activity started
21011	Activity ended normally

Archived

8.5 Developing the monitor model

To develop a monitor model for the implemented .fdl file, we must complete the following tasks:

- Install FDL to the monitor model utility for WebSphere MQ Workflow.
- Import the existing .fdl file.
- Generate a monitor model on the imported .fdl file.
- Augment the generated monitor model with more measures.
- Export the deployable EAR file.

8.5.1 Setting up the monitor model development environment

Before we install the FDL to monitor model utility, ensure that WebSphere Integration Developer and WebSphere Business Monitor development toolkit are installed. Because we develop the Visual model for our monitor model, ensure that the SVG viewer is also installed.

To install the FDL to monitor model utility:

1. Navigate to the WebSphere Business Monitor development toolkit package installation directory and find the **WMQWF** folder (Figure 8-30).

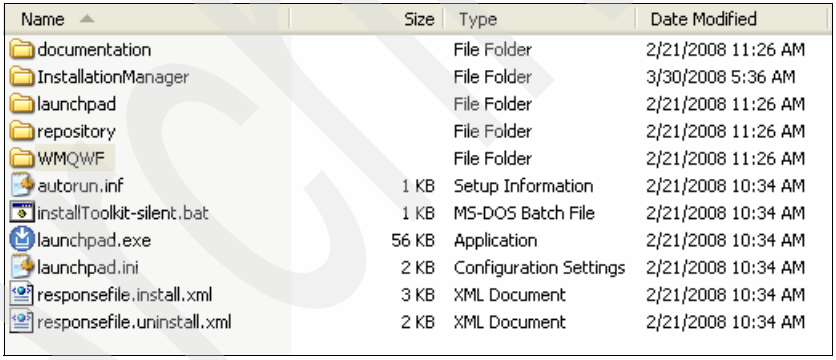


Figure 8-30 WMQWF folder

2. Navigate to the WMQWF folder and extract the contents of the compressed file (Figure 8-31).

Figure 8-31 Extracting the FDL_to_monitor_model_utility.zip file

3. Before we install the FDL to monitor model utility, modify feature.xml in <WebSphere Integration Developer installation root directory>\SDP70Shared\features\com.ibm.wbimonitor.runtime.feature_6.1.0.200802041700 to replace the plug-in version with `6.1.0.200802041700` (Figure 8-32). Doing this helps to overcome a known issue in the WebSphere Business Monitor 6.1.1 development toolkit.

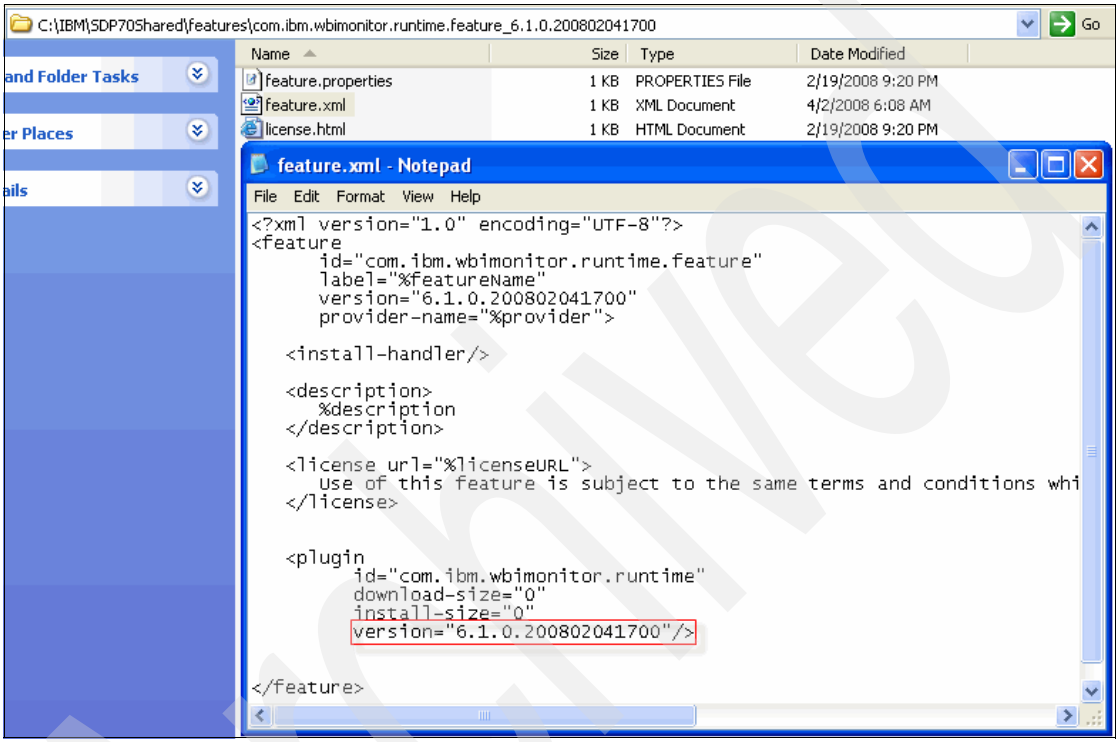


Figure 8-32 Modifying the plugin version

4. Start WebSphere Integration Developer.

Archive

5. From the menu bar, select **Help** → **Software Updates** → **Find and Install...** (Figure 8-33).

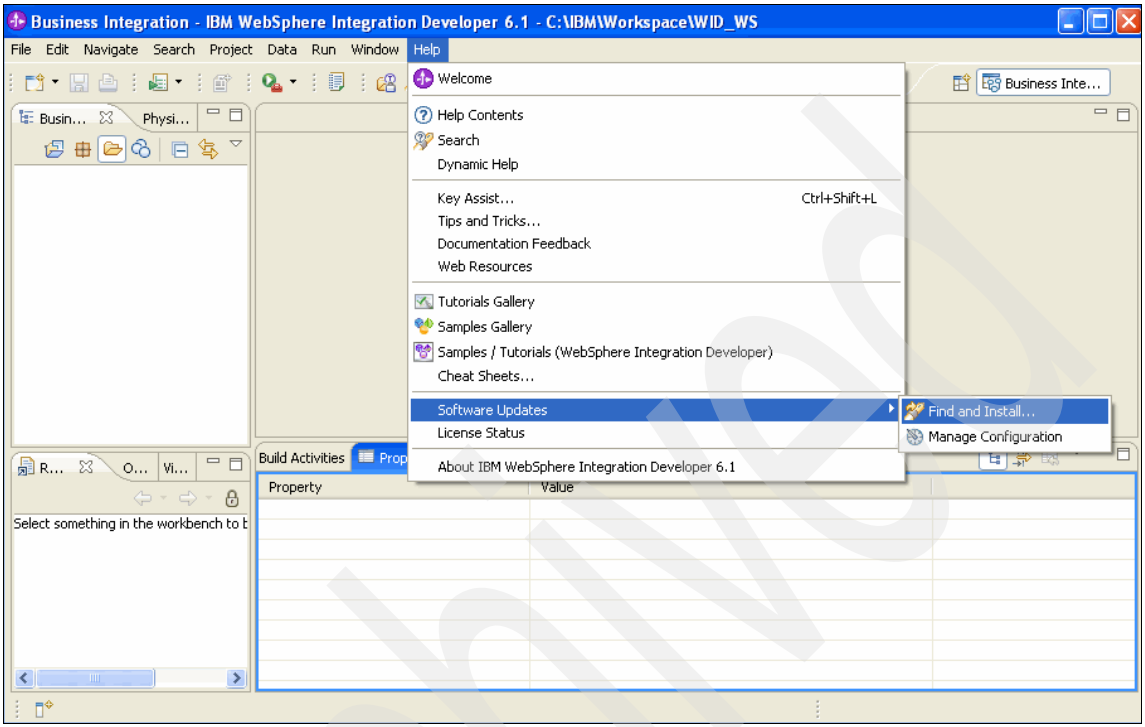


Figure 8-33 Finding and installing software updates

Archived

6. In the Install/Update window (Figure 8-34), select **Search for new features to install** and click **Next**.

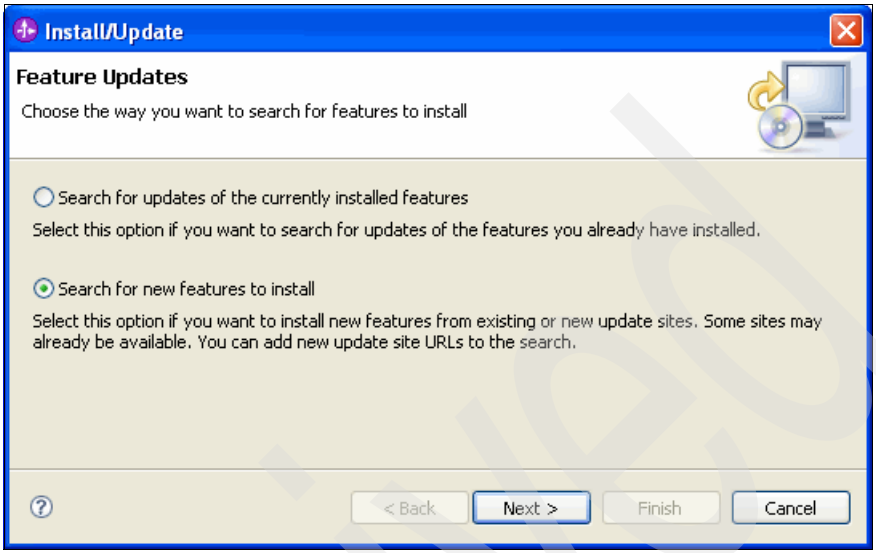


Figure 8-34 Selecting to install new features

Archived

7. In the Install – Update sites to visit window (Figure 8-35), click **New Local Site...**.

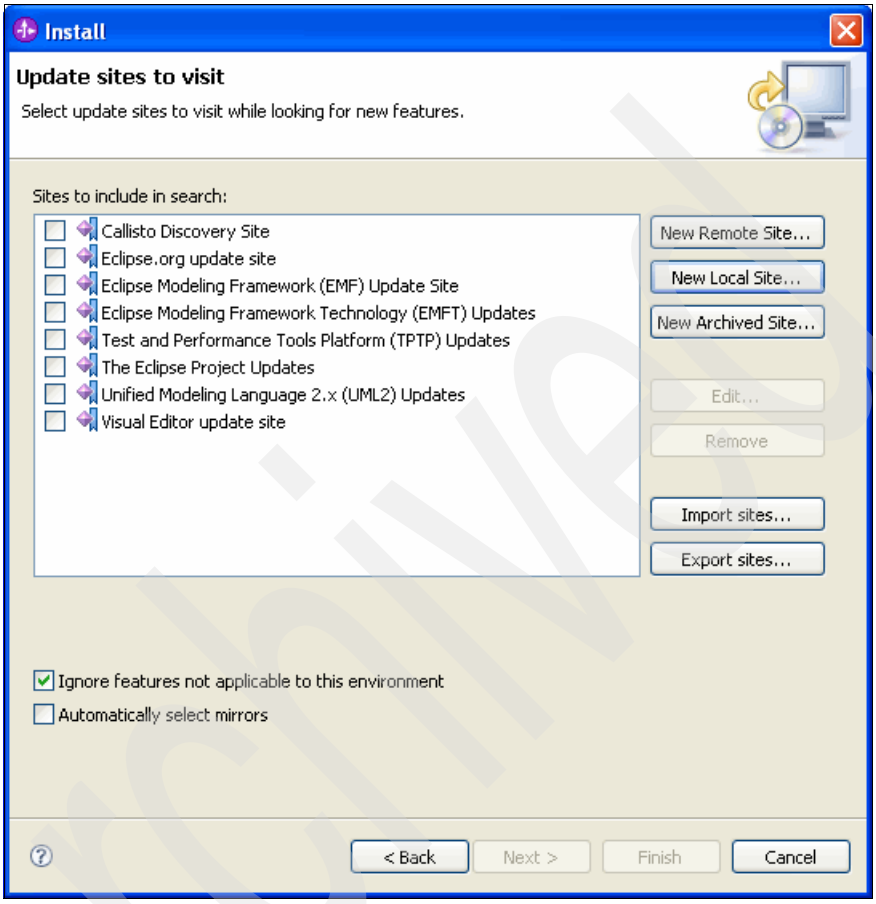


Figure 8-35 Install window

Archive

a. In the Browse for Folder window (Figure 8-36), locate the directory in which you extracted the FDL_to_monitor_model_utility.zip file and click **OK**.

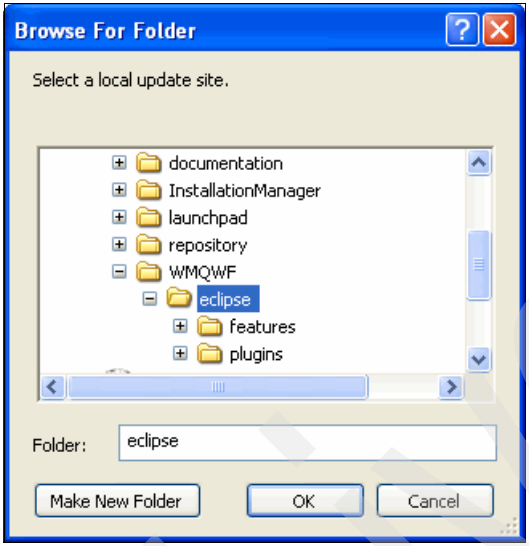


Figure 8-36 Locating the FDL_to_monitor_model_utility.zip file

b. In the Edit Local Site window (Figure 8-37), click **OK**.

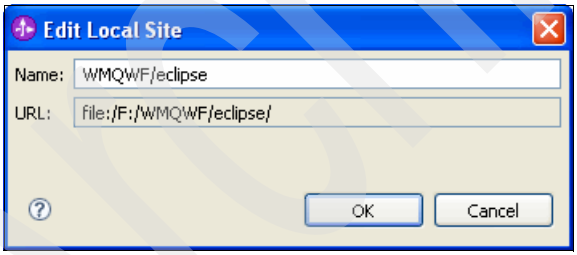


Figure 8-37 Setting the site name

Archived

c. Back in the Install – Update sites to visit window (Figure 8-38), make sure the newly created site is selected and click **Finish**.

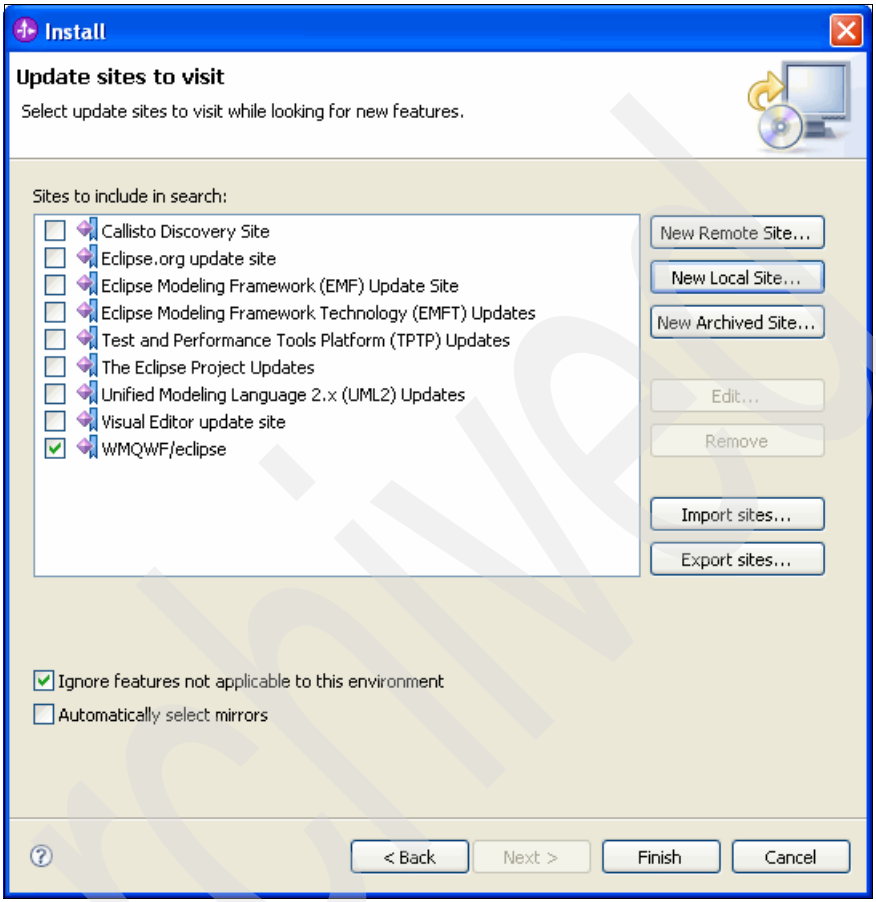


Figure 8-38 Selecting the newly created site

Archive

8. In the Updates window (Figure 8-39), select **WMQWF/eclipse** and click **Next**.

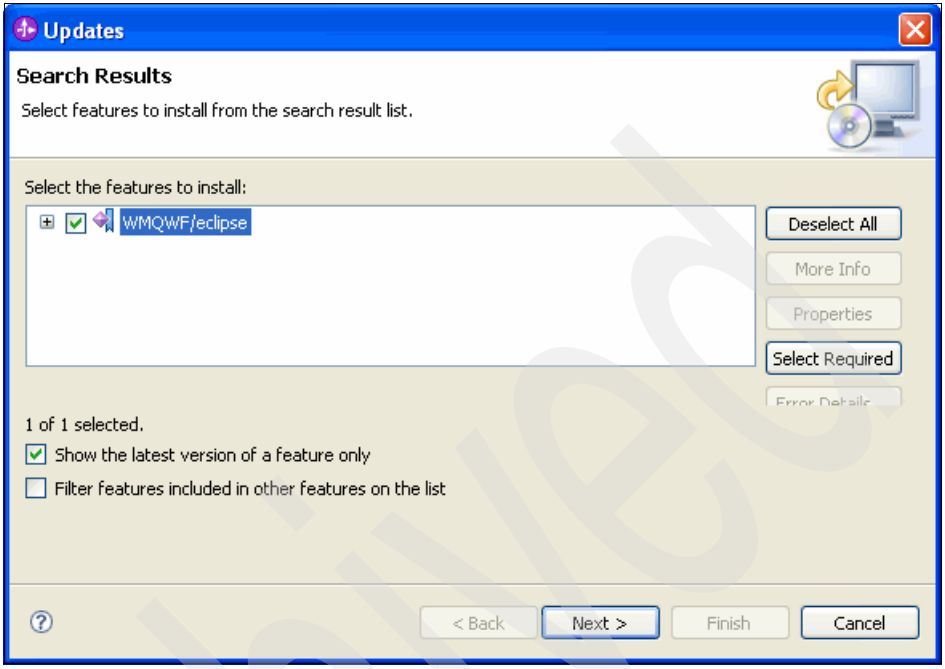


Figure 8-39 Selecting the newly created feature

Archived

9. In the Install – Feature License window (Figure 8-40), accept the license and click **Next**.

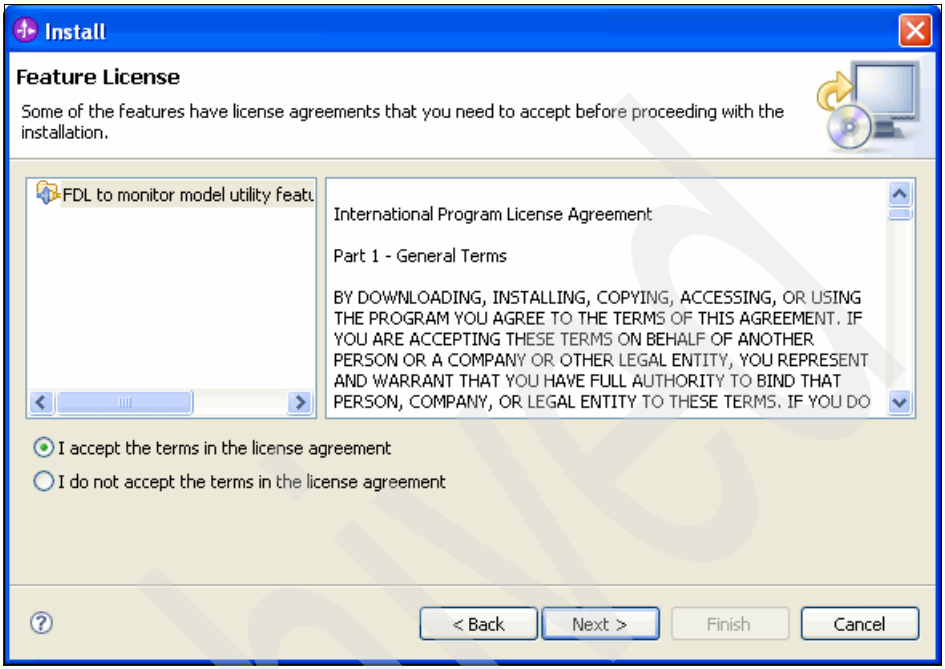


Figure 8-40 Accepting the license agreement

Archive

10. In the Install – Installation window (Figure 8-41), review the installation information and click **Finish**.

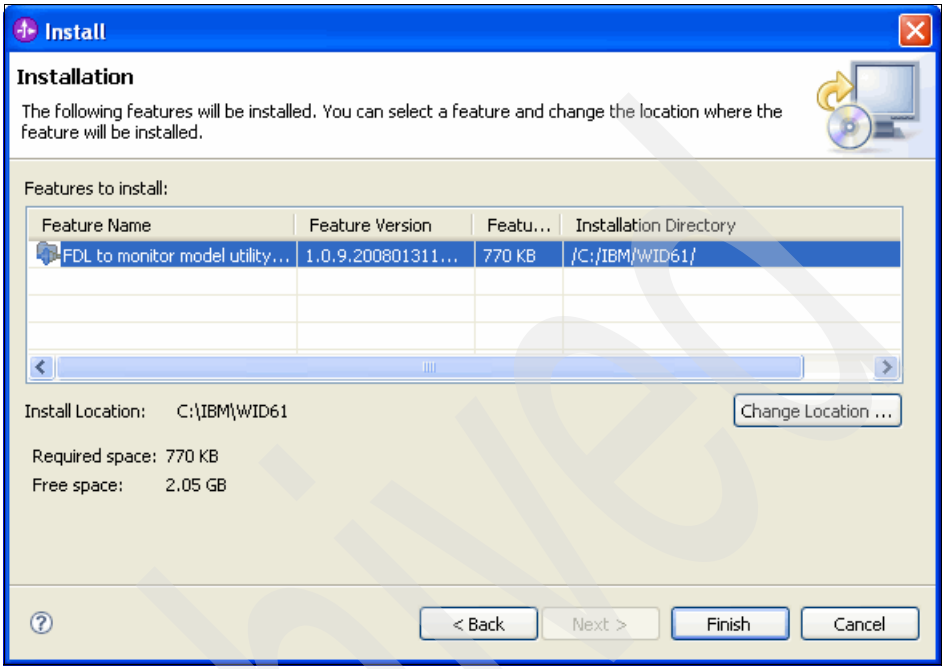


Figure 8-41 Summary of the installation information

Archive

11. In the Verification window (Figure 8-42), click **Install**.

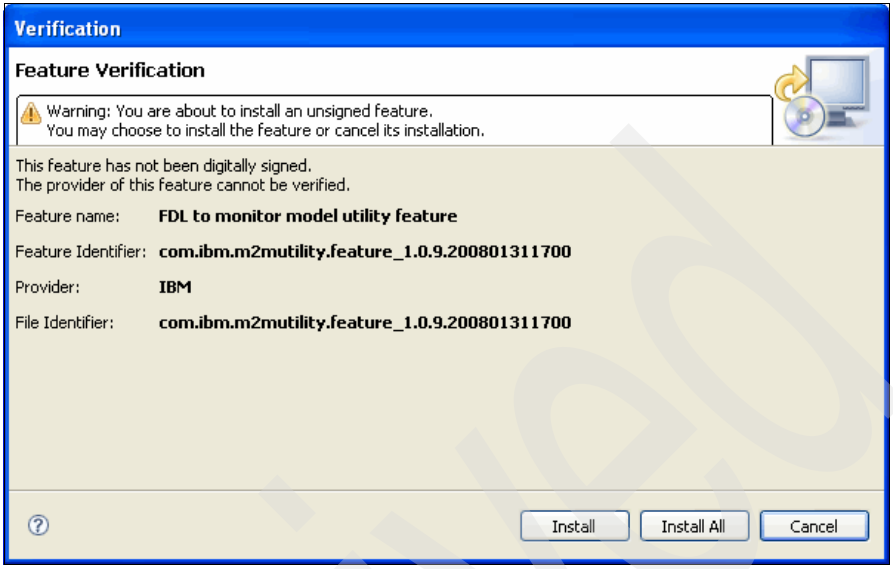


Figure 8-42 Verification window

12. After the installation completes, when you see the Install/Update message window (Figure 8-43), click **Yes** to restart WebSphere Integration Developer.

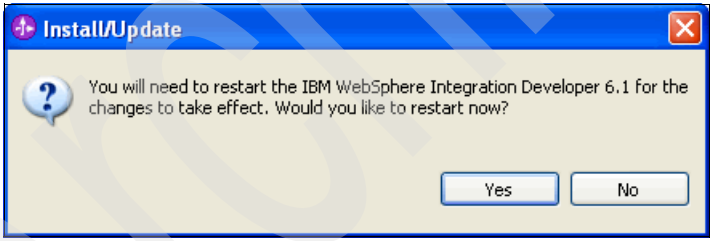


Figure 8-43 Clicking Yes to restart WebSphere Integration Developer

8.5.2 Developing the monitor model

Now WebSphere Integration Developer, WebSphere Business Monitor development toolkit, and the FDL to monitor model utility are installed. We need these applications to develop the monitor model based on an .fdl file.

Table 8-3 on page 520 shows the measures information we create for the monitor model. Then we can understand the kind of templates and events that we need to enable during the monitor model generation.

Table 8-3 Measures for the monitor model

Measure	Trigger	Expression	Dimension, aggregate measures or KPIs
OrderHandling Start Time Metric	OrderHandlingProcess Created	EventCreation	
OrderHandling End Time Metric	OrderHandlingProcess End	EventCreation	
City Metric	OrderHandlingProcess Start	Payload/Customer/City	Location dimension Level 2
Country Metric	OrderHandlingProcess Start	Payload/Customer/Country	Location dimension Level 1
Order Number Metric	OrderHandlingProcess Start	Payload/OrderNumber	
Total Price Metric	OrderHandlingProcess Start	Payload/TotalPrice	
Order Shipped Metric	ShippedOrder End CancelOrder End	100 or 0	Average KPI
Order Status Metric	ShippedOrder End CancelOrder End	“Shipped” or “Cancelled”	
Shipped Order Counter	ShipOrder End	Add One	Sum measure
Order Fulfillment Duration Stopwatch	OrderHandlingProcess Created ShippedOrder End CancelOrder End	Duration between Start and Stop	Average KPI
Late Order Shipped Event	Recurring wait time trigger with condition (Order Fulfillment Duration > 3 minutes)	Contains Order Number and Order Fulfillment Duration	

Archived

Generating the monitor model

To generate the monitor model:

1. In WebSphere Integration Developer, from the menu bar, click **File** → **Import...**
2. In the Import – Select window (Figure 8-44), select **FDL to Monitor Model Utility for WebSphere MQ Workflow** → **Import FDL File** and click **Next**.

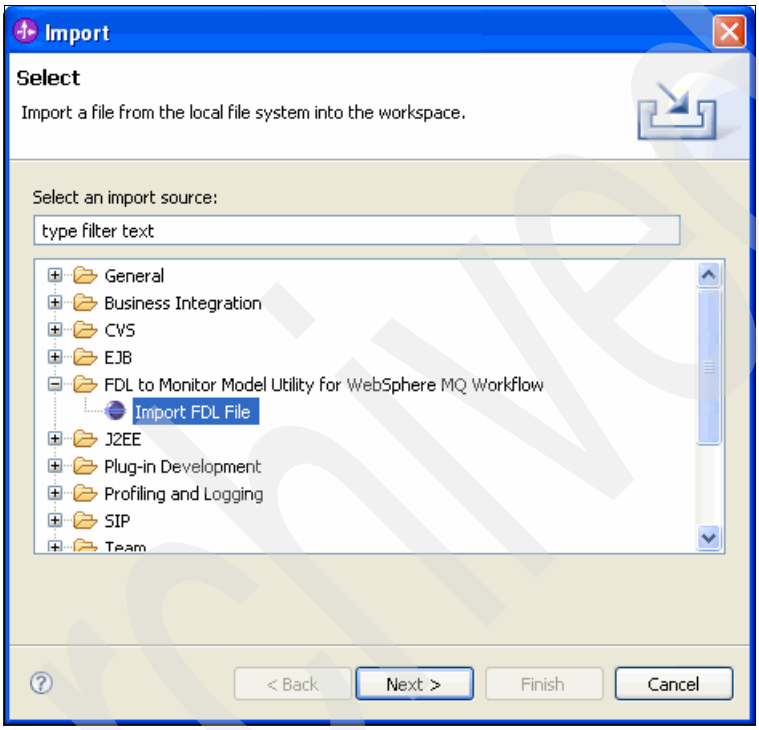


Figure 8-44 Selecting Import FDL File

Archived

3. In the FDL File Import Wizard window (Figure 8-45), for Select FDL File, click **Browse...** and locate the **OrderHandling.fdl** file. For Project name, select **OrderHandling**. Click **Finish**.

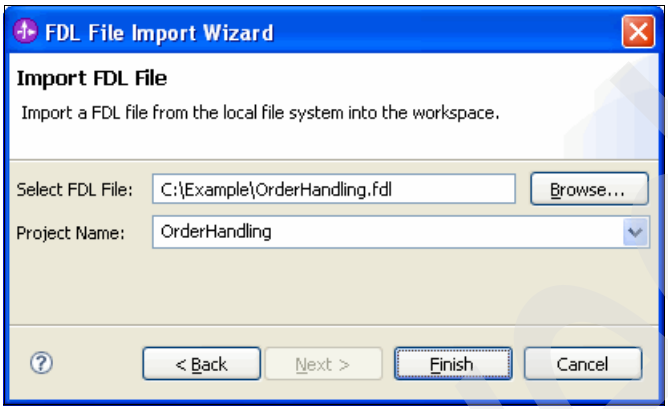


Figure 8-45 Selecting the .fdl file and setting the project name

4. If the current perspective is not the Java perspective, the Open Associated Perspective window (Figure 8-46) opens. Select **Remember my decision** and click **Yes** to switch to the Java perspective.

Figure 8-46 Selecting Yes to switch to the Java perspective

Archive

5. From the newly created OrderHandling project (Figure 8-47), expand **FDL**, right-click **OrderHandling.fdl** and select **Monitor** → **Generate Monitor Model for MQ Audit...**.

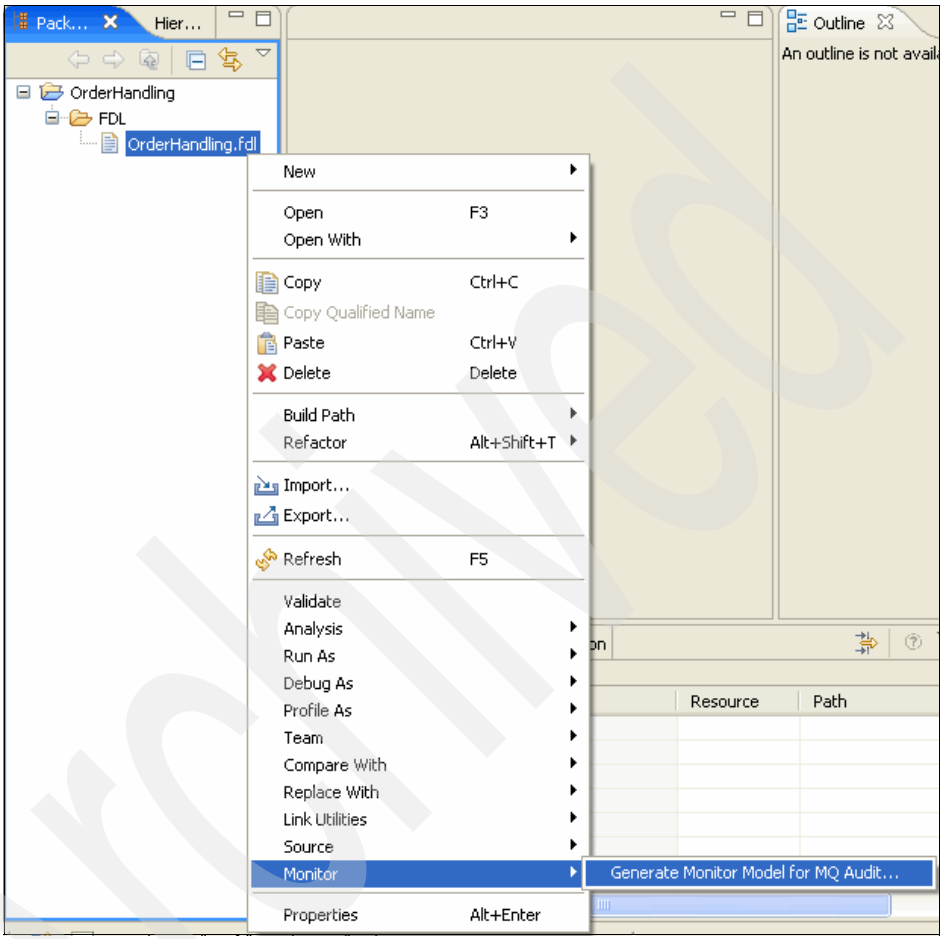


Figure 8-47 Selecting the option Generate Monitor Model for MQ Audit

Archived

6. In the Generate Monitor Model – Create a business monitoring project window (Figure 8-49):
   a. For Target monitor project, click **New Project...**
   b. In the New Business Monitoring Project window (Figure 8-48), for Project name, type `OrderHandlingBM` and click **Finish**.

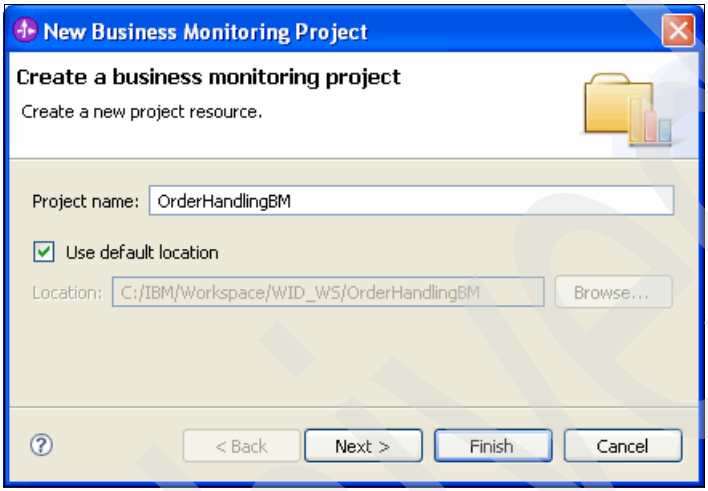


Figure 8-48 Setting the new business monitoring project name

   c. Back in the Generate Monitor Model window, for Target monitor model name, type `OrderHandlingMM`. Click **Next**.

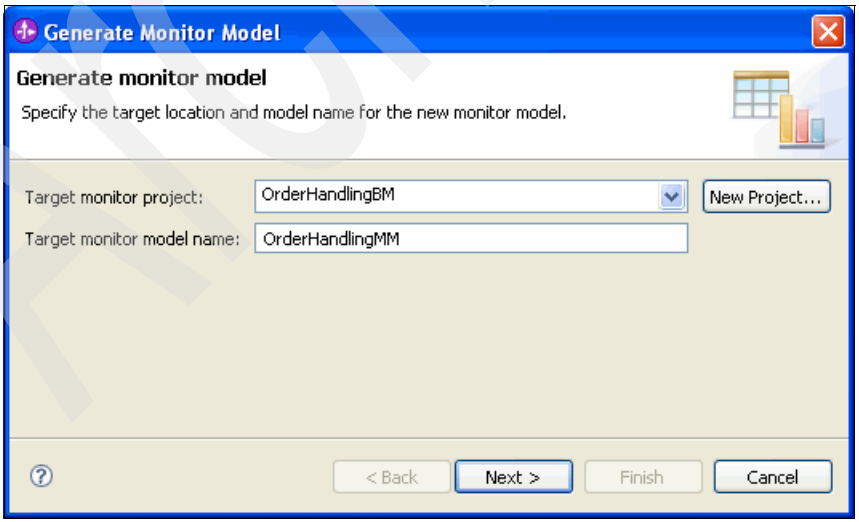


Figure 8-49 Setting the monitor model name

7. In the Generate Monitor Model – Choose what to monitor window (Figure 8-50):

   a. Choose the monitoring templates and events. Select the **Limit my selection of events and templates based on the events that have been turned on in the application** check box. By selecting this option, only those templates that are available for the current .fdl file are displayed on the Monitoring Template page.

   b. Under Event Source, expand **WMQWF.OrderHandling** → **OrderHandling**.

   c. Click the **Monitoring Templates** tab. Select **Start Time** and **End Time**.

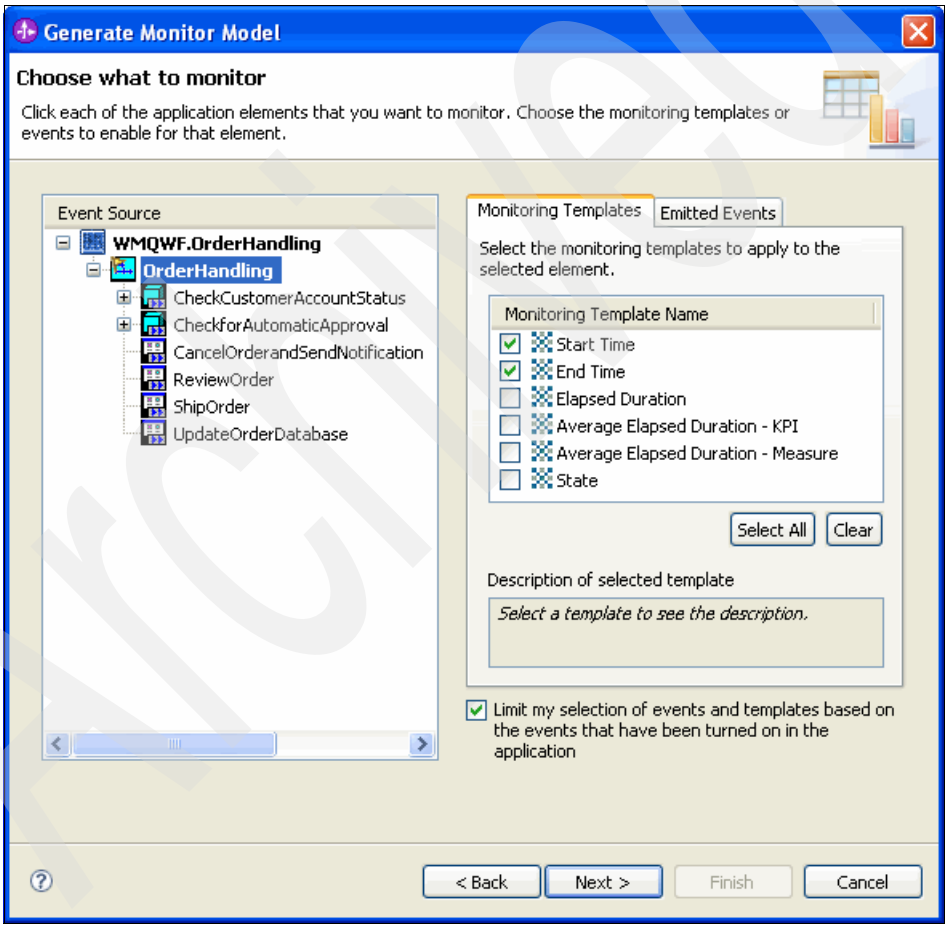


Figure 8-50 Choosing the monitoring template for the OrderHandling process

Archived

d. Under Event Source, click **CancelOrderandSendNotification**. Click the **Emitted Events** tab. Select **CancelOrderandSendNotification_Activity _Ended_Normally**. See Figure 8-51.

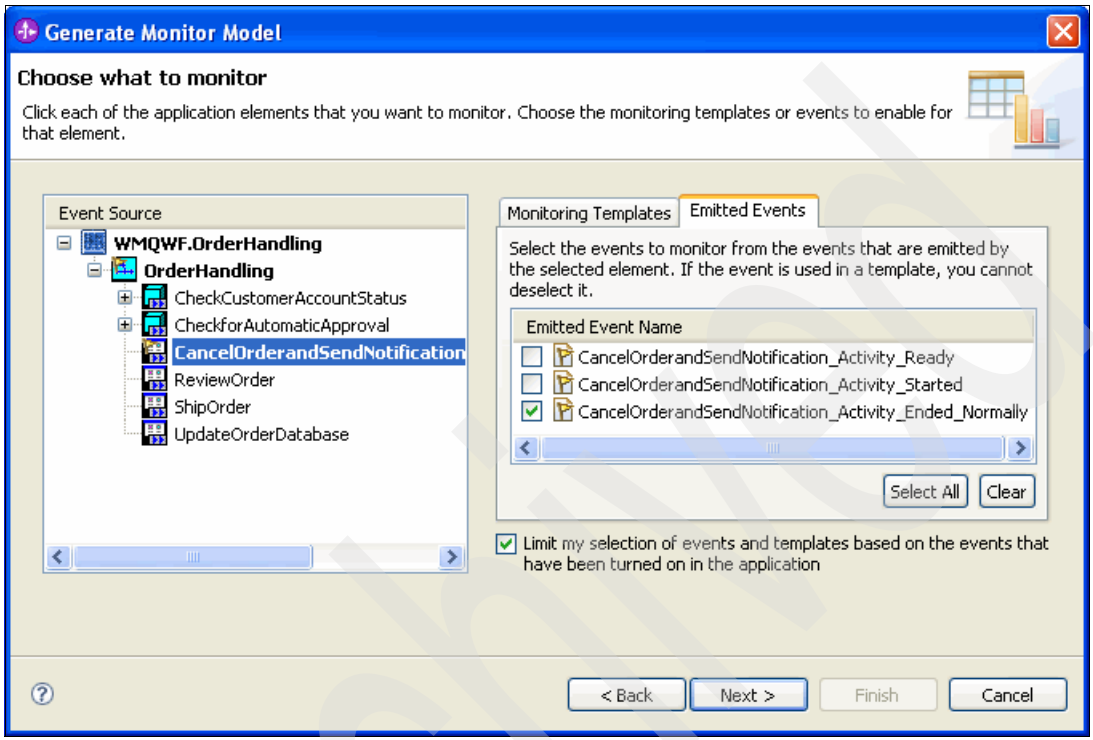


Figure 8-51 Choosing the emitted event for the CancelOrderandSendNotification activity

Archive

e. Under Event Source, click **ShipOrder**. On the Emitted Events page, select **ShipOrder_Activity_Ended_Normally**. See Figure 8-52.

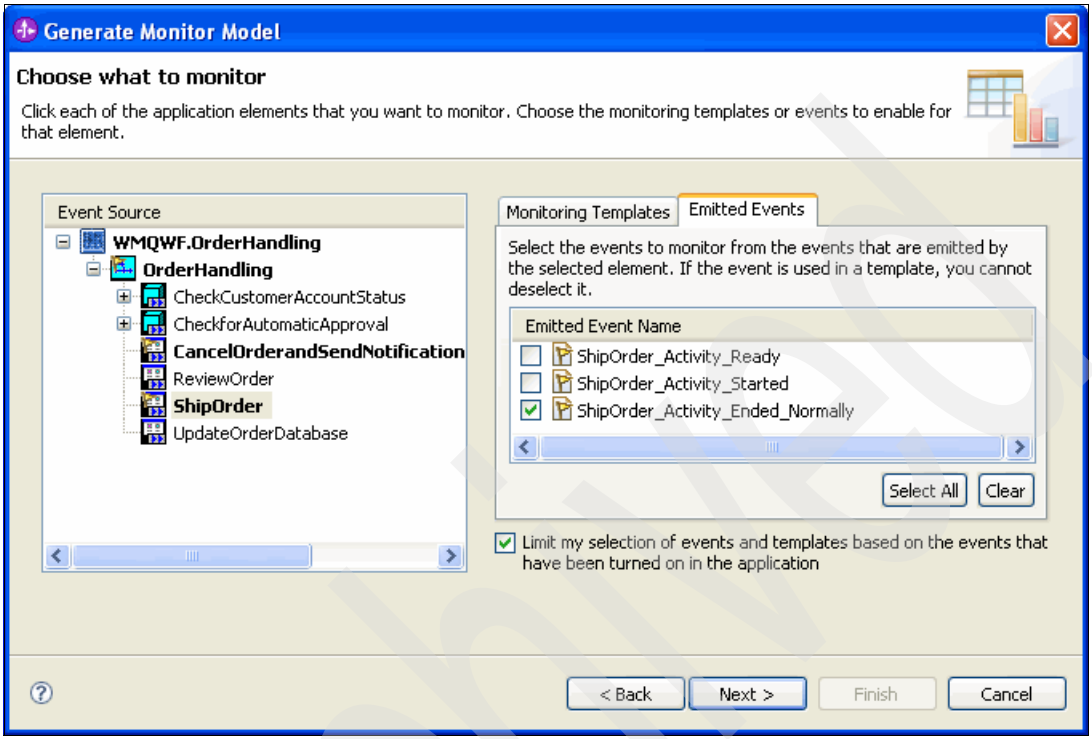


Figure 8-52 Choosing the emitted event for the ShipOrder activity

f. Click **Next**.

Archived

8. In the Generate Monitor Model – Choose how to monitor window (Figure 8-53), for CancelOrderandSendNotification and ShipOrder, change Implementation to **Event group**.

   We make this change so that we do not have measures under CancelOrderandSendNotification and ShipOrder activities. Therefore, we do not need monitoring context implementation for them. Monitoring contexts introduce overhead in the form of keys, cubes, and so on. Event groups are simply containers to group the related inbound events, which are purely the visual construct.

   Click **Next**.

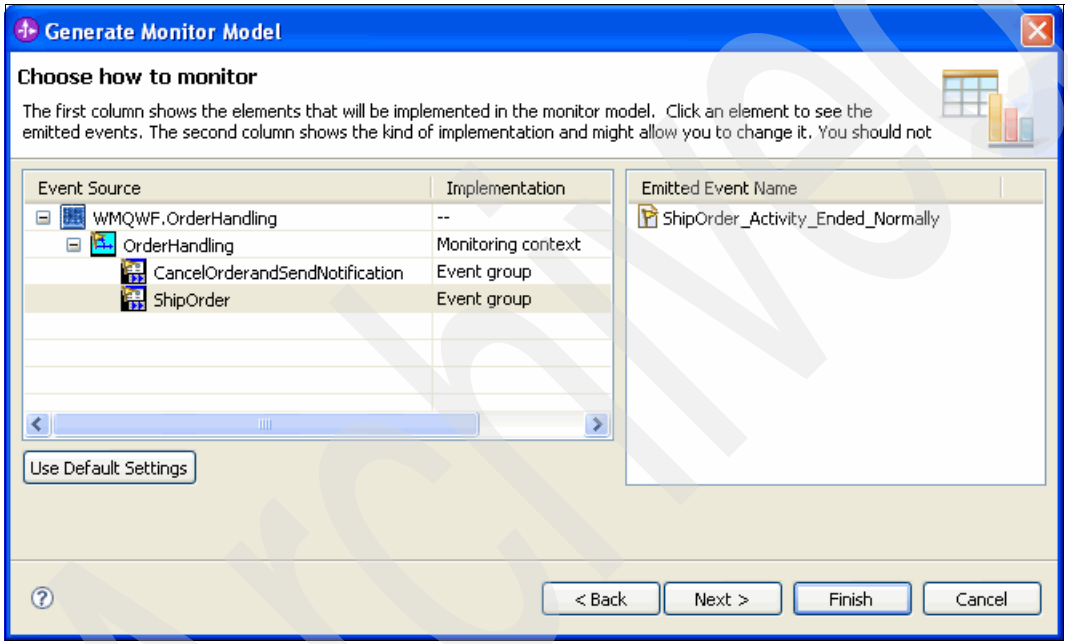


Figure 8-53 Changing the activities' implementation to be an event group

Archived

9. In the Generate Monitor Model – Preview the monitor model window (Figure 8-54), review the preview information for the monitor model. Click **Finish**.

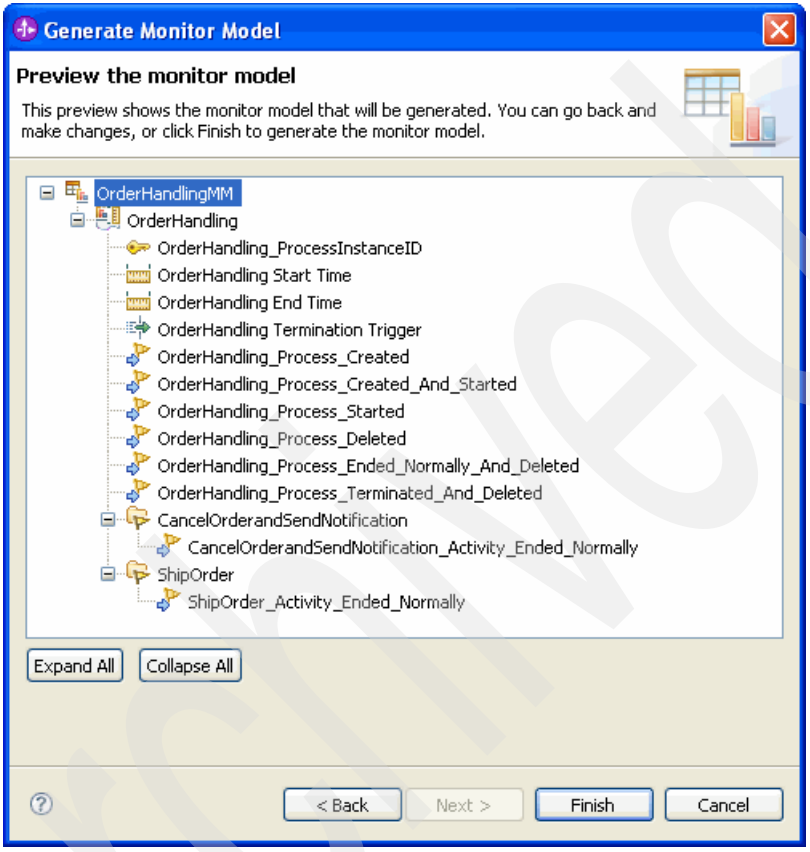


Figure 8-54 Previewing the monitor model

Archived

10. In the Switch perspectives message window (Figure 8-55), select **Remember my decision** and click **Yes** to switch to the Business Monitoring perspective.

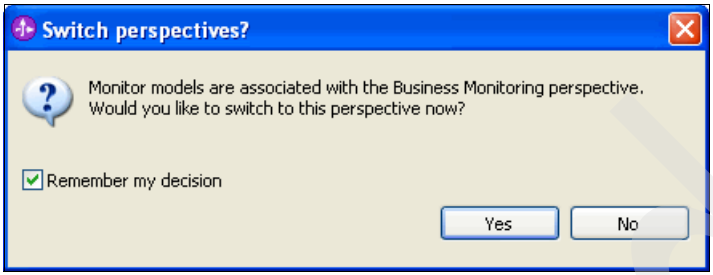


Figure 8-55 Switching to the Business Monitoring perspective

11. In the Launch Get Started message window (Figure 8-56), click **No**.

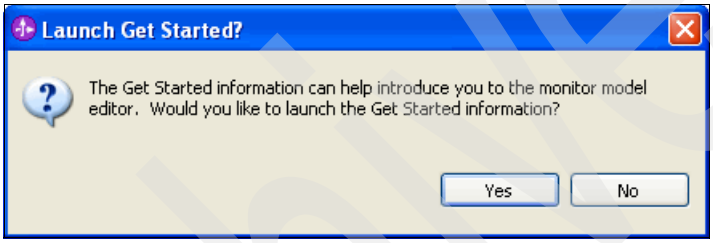


Figure 8-56 Selecting No to skip the Get Started information

Archived

The monitor project and monitor model is generated as shown in Figure 8-57. Based on this initial model, we add more measures in the following sections.

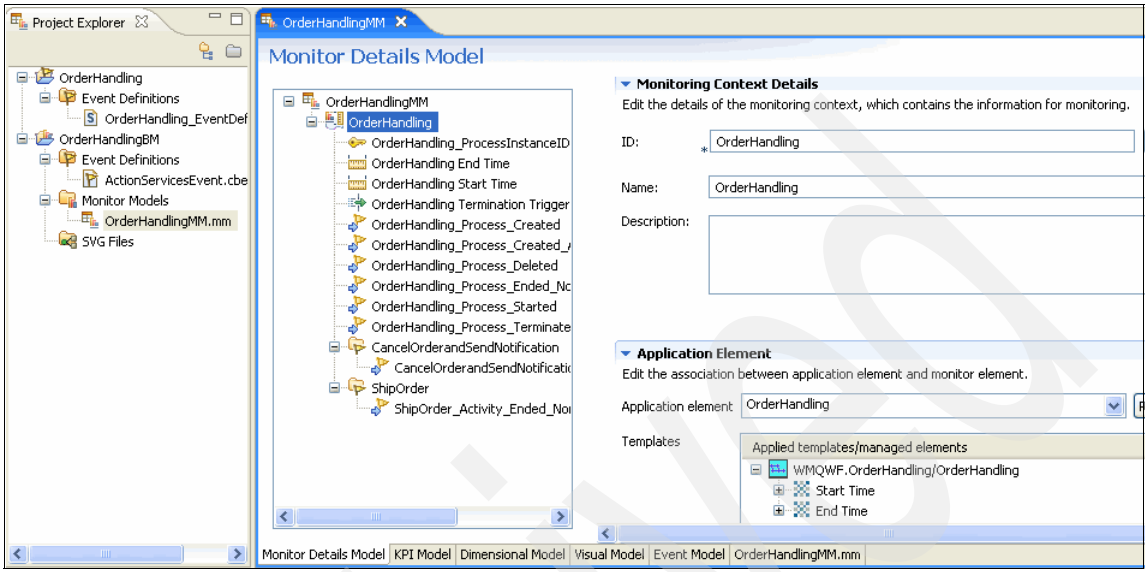


Figure 8-57 Generated monitor project and monitor model

Defining the monitor details model

In the following steps, we augment the monitor details model. Most of the monitor model information is defined in the monitor details model, for example the monitoring context and its associated metrics, keys, counters, stopwatches, triggers, and events. The keys and some triggers and metrics are generated in the initial monitor model. Therefore we only add the new elements listed in Table 8-4.

Table 8-4 Elements added to the monitor details model

Element type	Element name
Metric	City, Country, Order Number, Total Price, Order Shipped, Order Status
Trigger	Order Cancelled Trigger, Order Cancelled Trigger, Late Order Fulfillment Event
Stopwatch	Order Fulfillment Duration
Counter	Shipped Order
Outbound event	Late Order Shipped Event

To define the monitor details model:

1. Right-click **OrderHandling** and click **New** → **Metric** to add a metric (Figure 8-58).

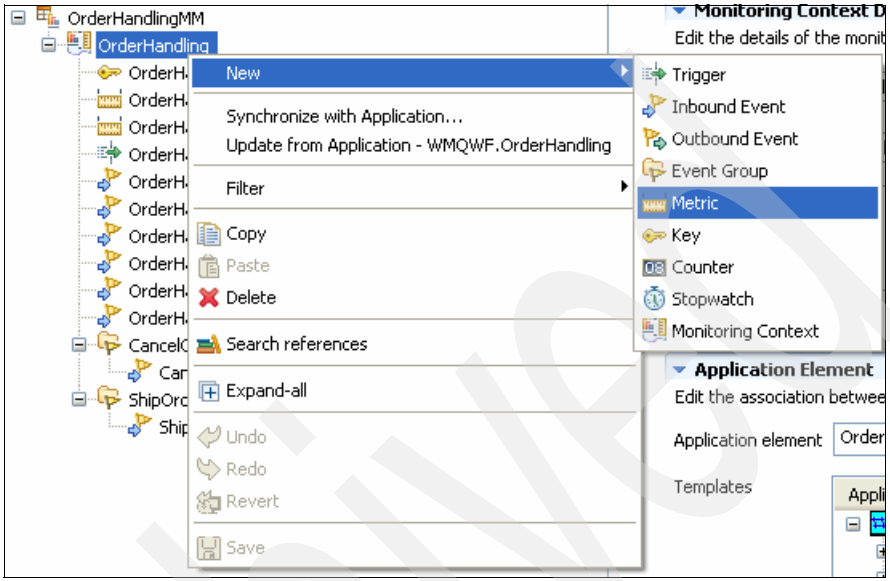


Figure 8-58 Adding a metric

2. In the Create New Metric window (Figure 8-59), for Name, type `City` and click **OK**.

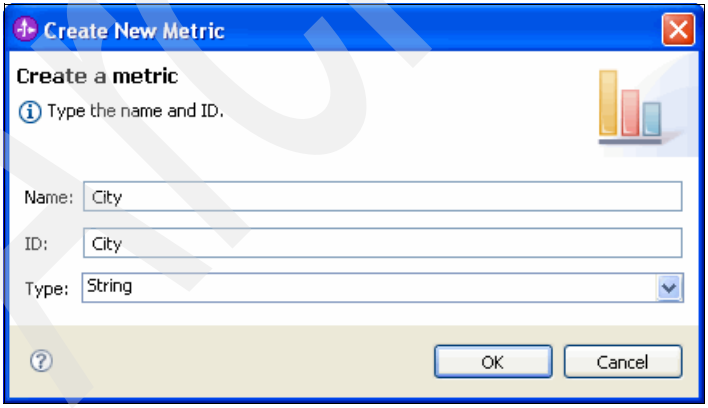


Figure 8-59 Setting the metric name

Archive

3. Because we use the City metric as a dimension, on the Monitor Details Model page (Figure 8-60), select **A value is required for this metric**. For Default value, type `'Unknown'`. Under Metric Value Expressions, click **Add** and then click the **...** button on the right in the Expression cell.

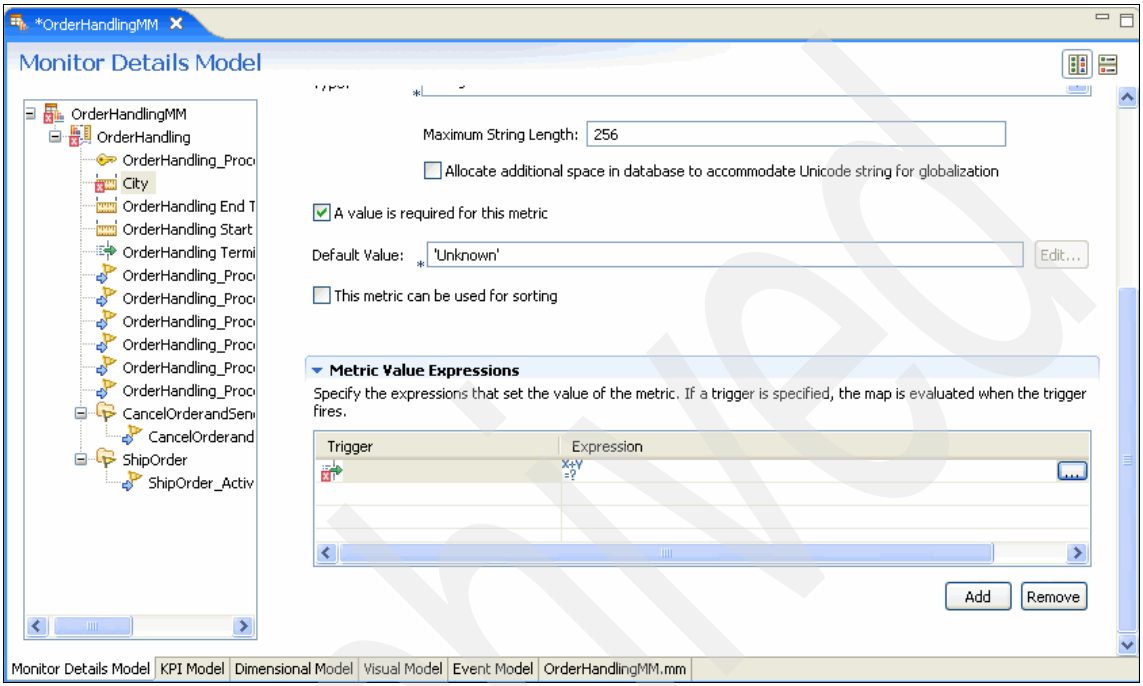


Figure 8-60 Setting the default value for City metric

4. In the Expression Dialog (Figure 8-61), click the blank area and press Ctrl+Spacebar to activate the Content Assist.

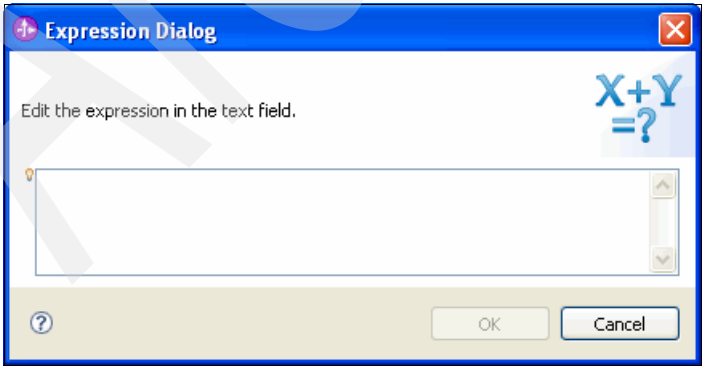


Figure 8-61 Expression Dialog

5. In the Content Assist (Figure 8-62), expand **OrderHandlingMM** → **OrderHandling** → **OrderHandling_Process_Created_And_Started** → **ApplicationData : Order** → **Customer : CustomerInformation** and double-click **City : string**. Click **OK**.

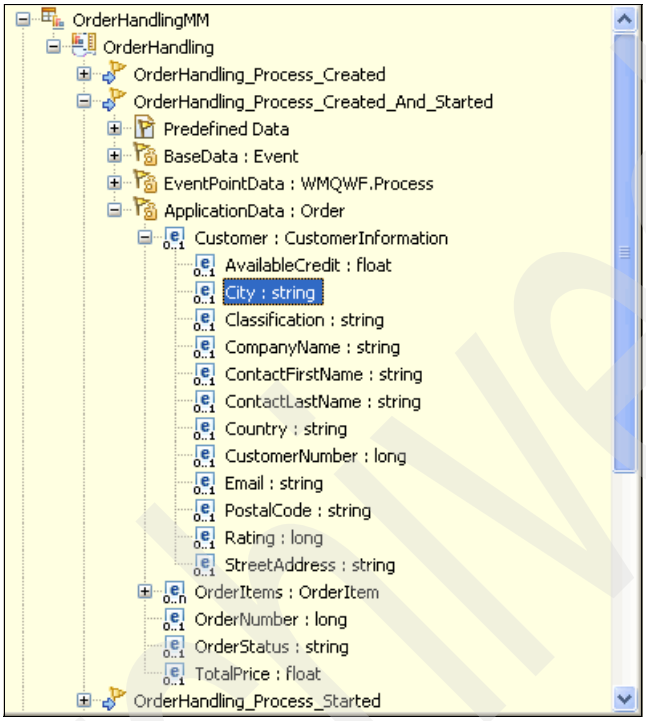


Figure 8-62 Setting a value for City metric

6. In the same way set one more expression for the City metric using the OrderHandling_Process_Started event (Figure 8-63).

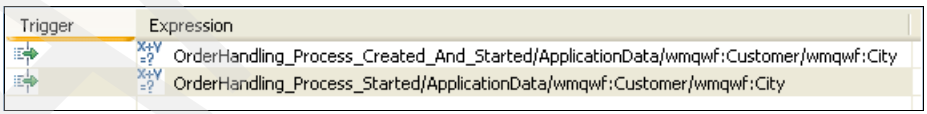


Figure 8-63 Setting one more expression for the City metric

Archive

We set this second expression so that we can start the WebSphere MQ Workflow process in either of the following ways:

- Right-click the process template and select **Create and Start Instance**. In this way, the Process_Created_And_Started event is emitted and carries the customer information.
- Right-click the process template and select **Create Instance**. Then right-click the newly created process instance and select **Start**. In this way, the Process_Created and Process_Started events are emitted and only the Process_Started event carries the customer information.

7. Add a Country metric with configurations and expressions shown in Figure 8-64:
   a. Select **A value is required for this metric**.
   b. For Default Value, type `'Unknown'`.
   c. For Expression, type `OrderHandling_Process_Created_And_Started/ApplicationData/wmqwf:Customer/wmqwf:Country` and `OrderHandling_Process_Started/ApplicationData/wmqwf:Customer/wmqwf:Country`.

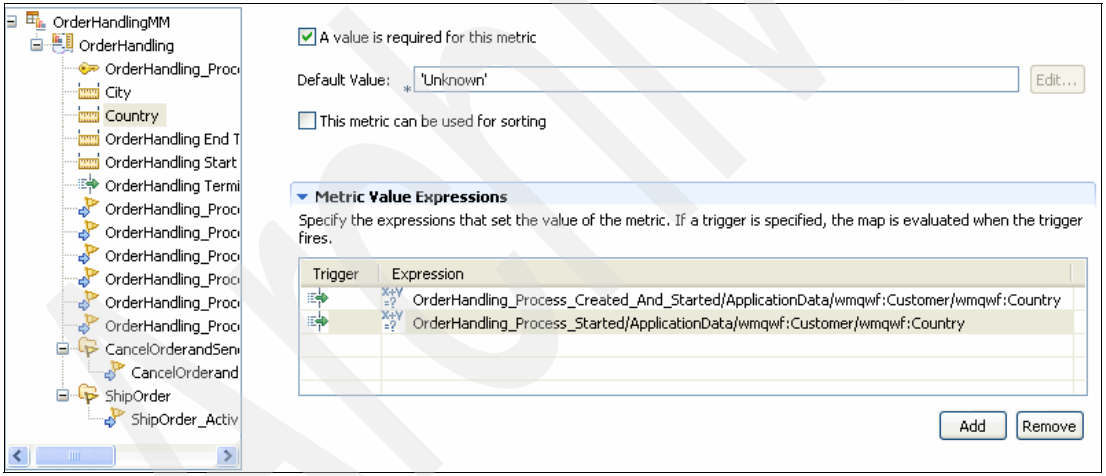


Figure 8-64 Adding the Country metric

8. Add an Order Number metric with configurations and expressions shown in Figure 8-65. For Type, select **Integer**. Under Metric Value Expression, type `OrderHandling_Process_Created_And_Started/ApplicationData/wmqwf:OrderNumber` and `OrderHandling_Process_Started/ApplicationData/wmqwf:OrderNumber`.

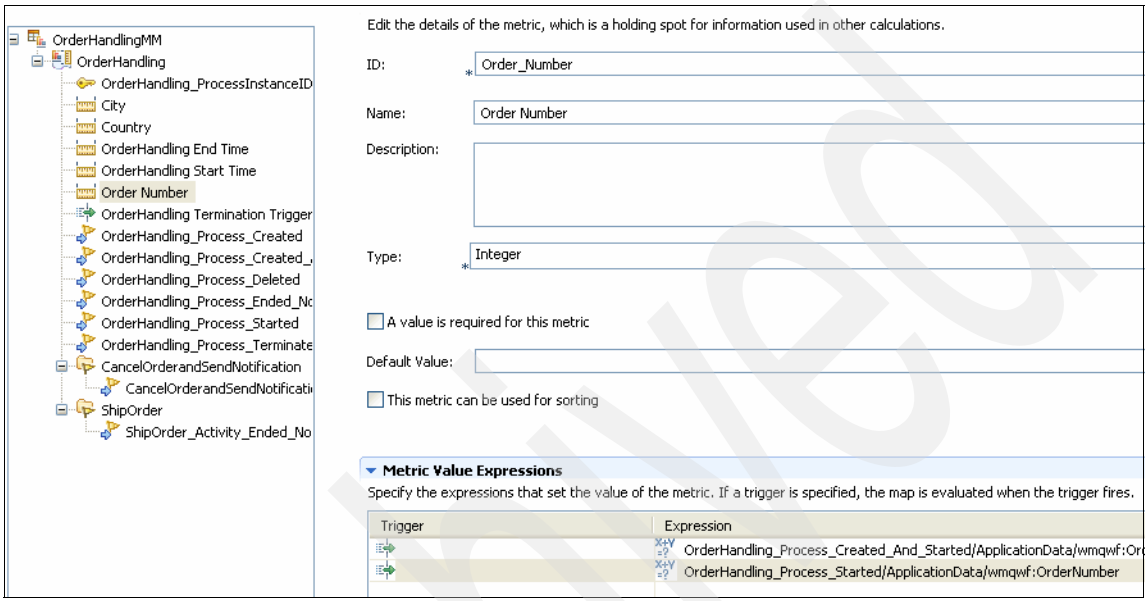


Figure 8-65 Adding the Order Number metric

Archived

9. Add a Total Price metric with configurations and expressions shown in Figure 8-66. For Type, select **Decimal**. Under Metric Value Expression, type `OrderHandling_Process_Created_And_Started/ApplicationData/wmqwf:TotalPrice` and `OrderHandling_Process_Started/ApplicationData/wmqwf:TotalPrice`.

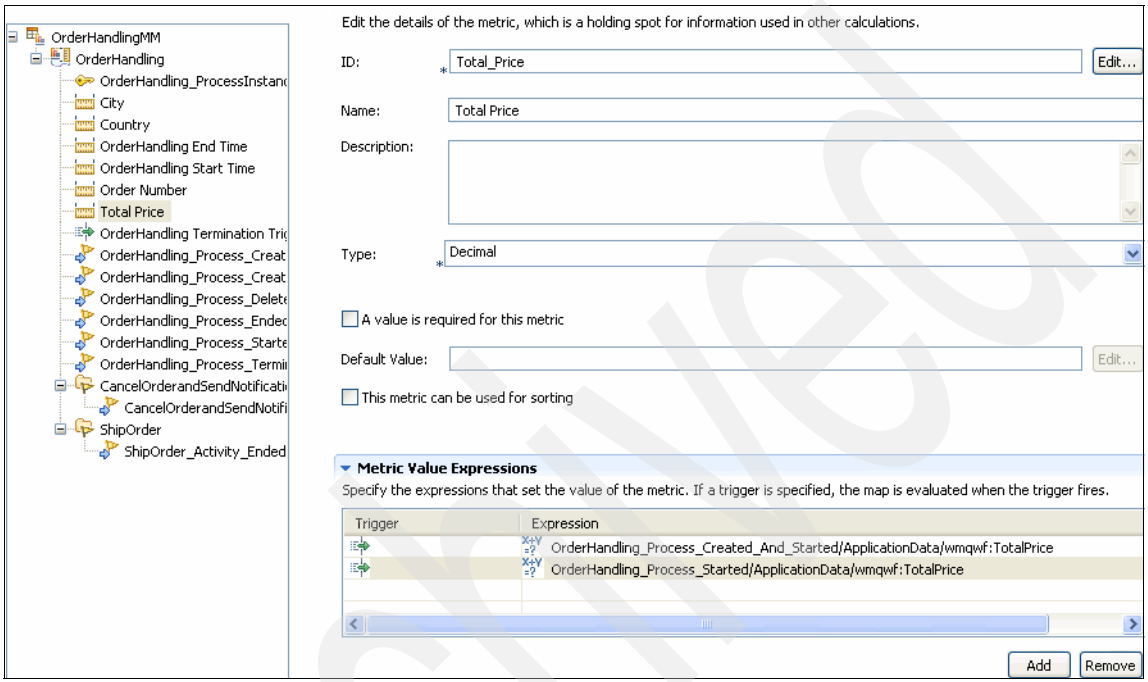


Figure 8-66 Adding the Total Price metric

10. Save your work.

Archived

11. Right-click **OrderHandling** and select **New** → **Trigger** to add a trigger (Figure 8-67).

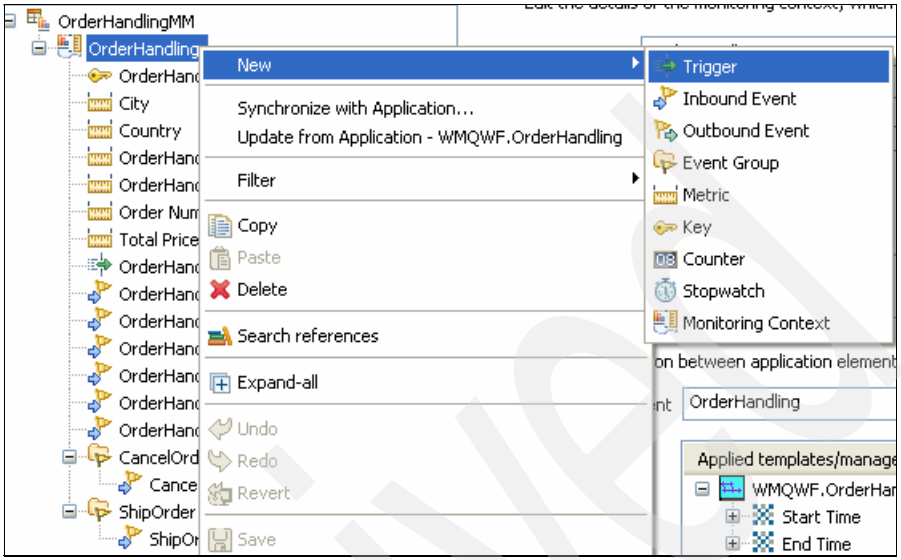


Figure 8-67 Adding a trigger

12. In the Create New Trigger window (Figure 8-68), for Name, type `Order Shipped Trigger` and click **OK**.

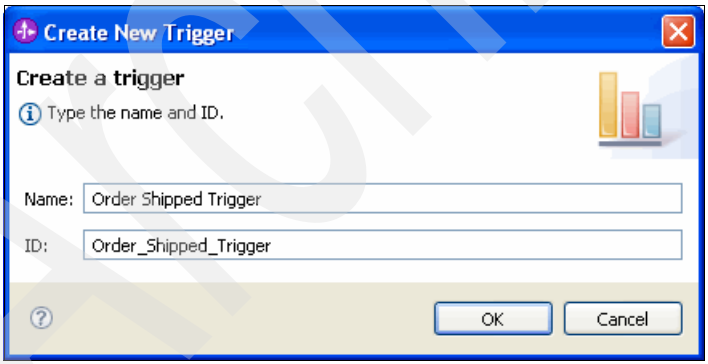


Figure 8-68 Setting the trigger name

Archived

13. Under Trigger Sources (Figure 8-69), for the newly created trigger, click the **Add** button.

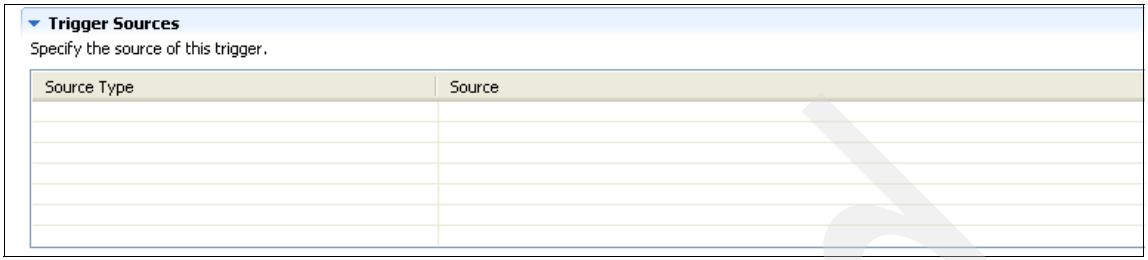


Figure 8-69 Trigger Sources for a trigger

14. In the Select Trigger Source window (Figure 8-70), click **ShipOrder_Activity_Ended_Normally** and click **OK**.

Figure 8-70 Selecting a trigger source for Order Shipped Trigger

Archive

15. In similar way, add an Order Cancelled Trigger with source `CancelOrderandSendNotification_Activity_Ended_Normally` as shown in Figure 8-71.

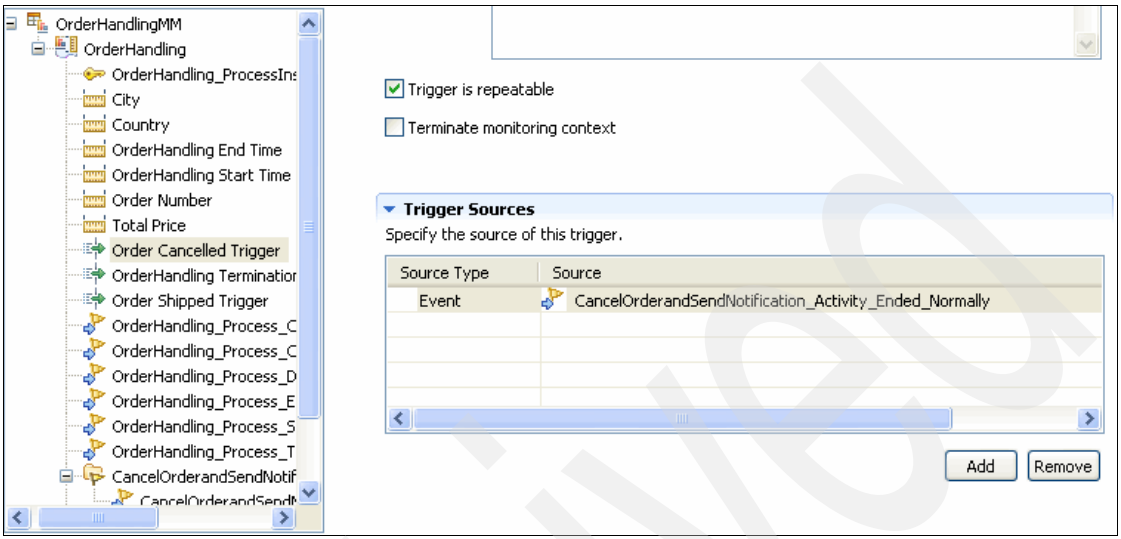


Figure 8-71 Adding the Order Cancelled Trigger

16. Save your work.

Archived

17. Add the Order Shipped metric with the configuration as follows (Figure 8-72). Each expression is triggered by the newly created trigger. The average value of this metric is exactly the percentage of shipped orders that are defined as KPIs later.

    a. For Metric Name, type `Order Shipped`.
    b. For Type, select **Integer**.
    c. For Expression for the Order Cancelled Trigger, type `0`.
    d. For Expression for the Order Shipped Trigger, type `100`.

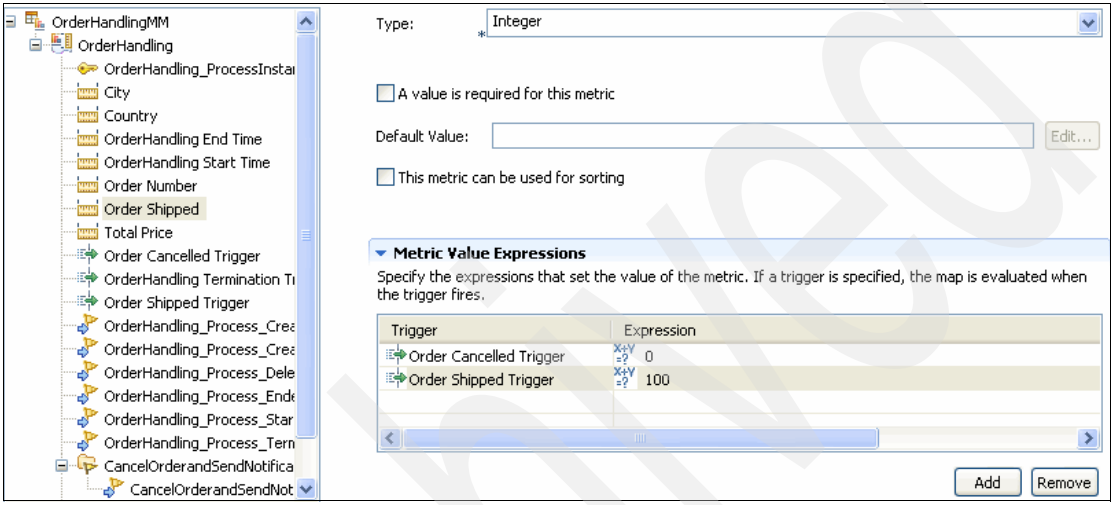


Figure 8-72 Adding the Order Shipped metric

Archived

18. Add the Order Status metric with configuration as follows (Figure 8-73). This metric is used in the visual model later.

    a. For Metric Name, type `Order Status`.
    b. For Type, select **String**.
    c. Select **A value is required for this metric**.
    d. For Default value, type `'New'`.
    e. For Expression for the Order Shipped Trigger, type `'Shipped'`.
    f. For Expression for the Order Cancelled Trigger, type `'Cancelled'`.

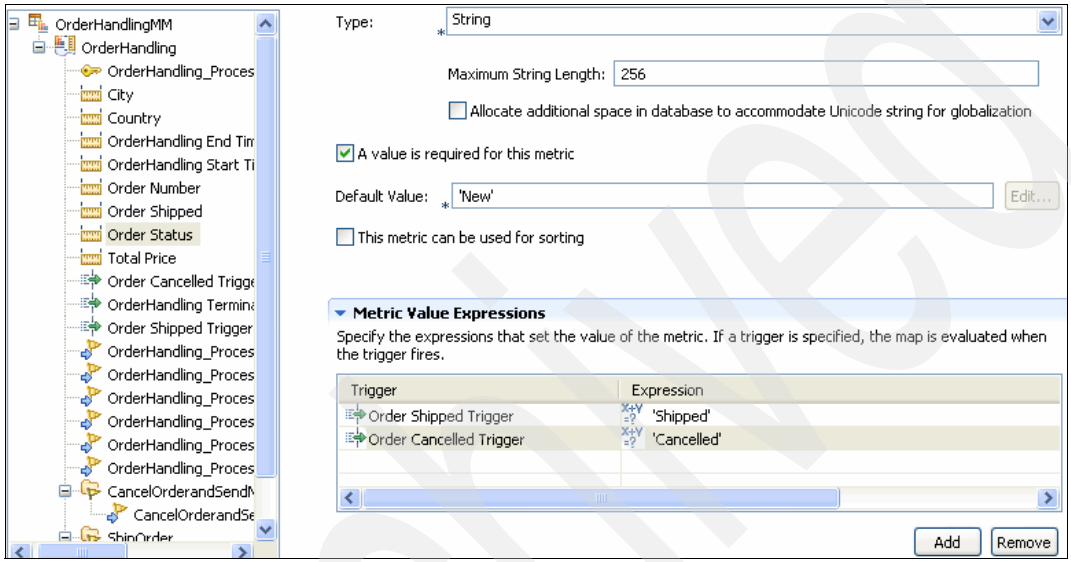


Figure 8-73 Adding the Order Status metric

19. Save your work.

Archived

20. Right-click **OrderHandling** and select **New** → **Counter** (Figure 8-74) to add a counter.

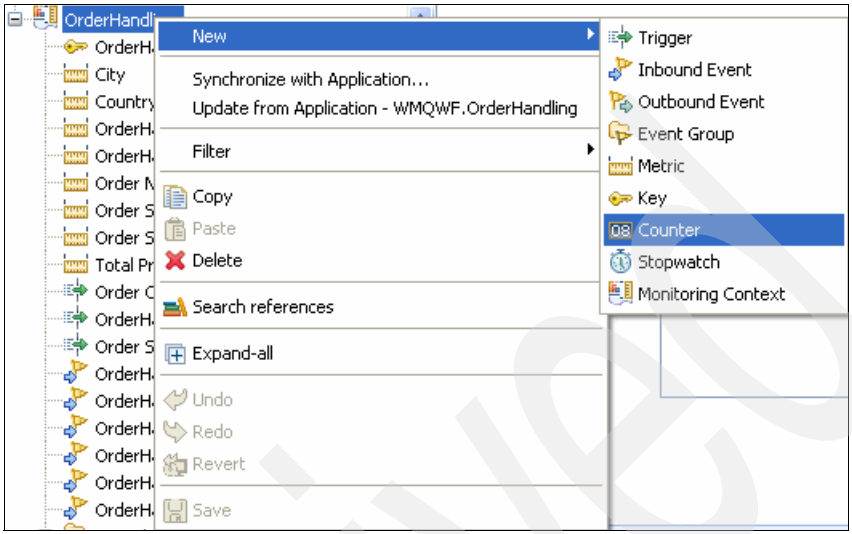


Figure 8-74 Adding a counter

21. In the Create New Counter window (Figure 8-75), set Name to `Shipped Order Counter` and click **OK**.

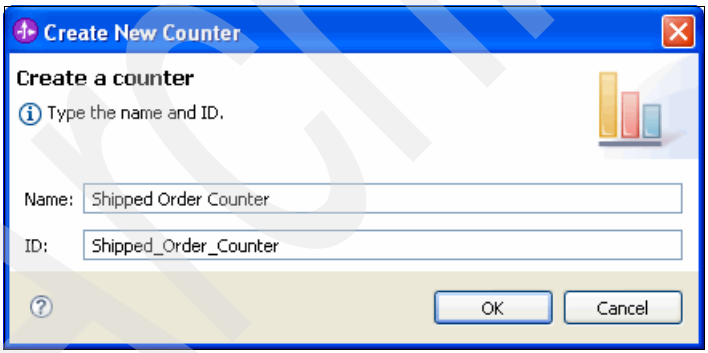


Figure 8-75 Setting the counter name

Archived

22. Under Counter Controls (Figure 8-76), for the newly created counter, click the **Add** button.

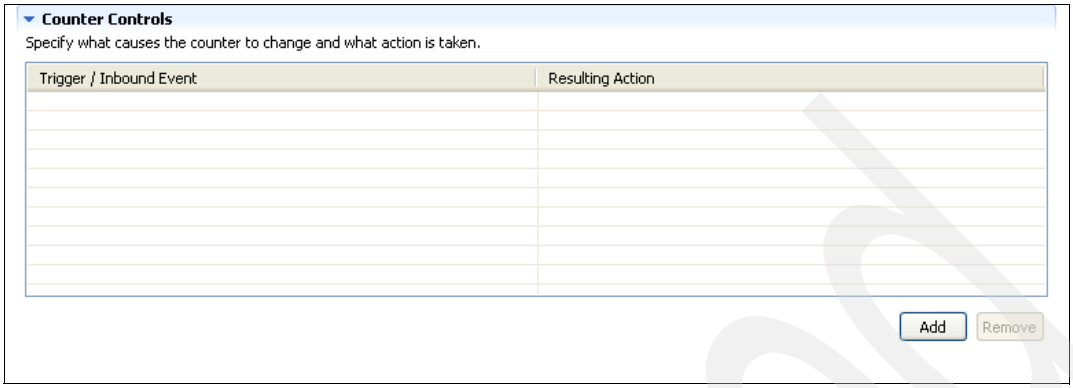


Figure 8-76 Counter Controls area for a counter

23. In the Select Trigger or Inbound Event window (Figure 8-77), click **ShipOrder_Activity_Ended_Normally** and click **OK**.

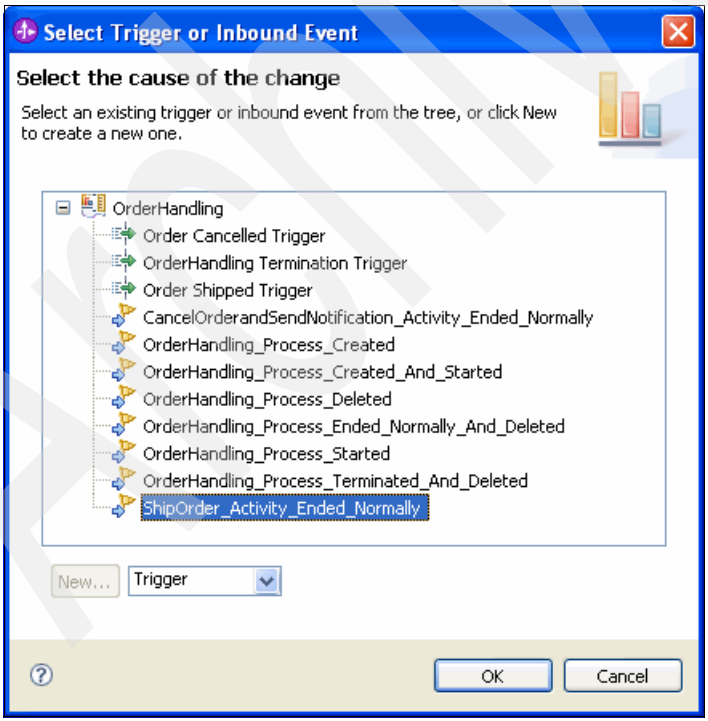


Figure 8-77 Selecting the inbound event for Shipped Order Counter

24. Under Counter Controls (Figure 8-78), ensure that Resulting Action for the selected inbound event is **Add One** so that after the ShipOrder activity completes successfully, the counter has a value of 1. This counter is used later in the dimensional model to calculate the sum of shipped orders.

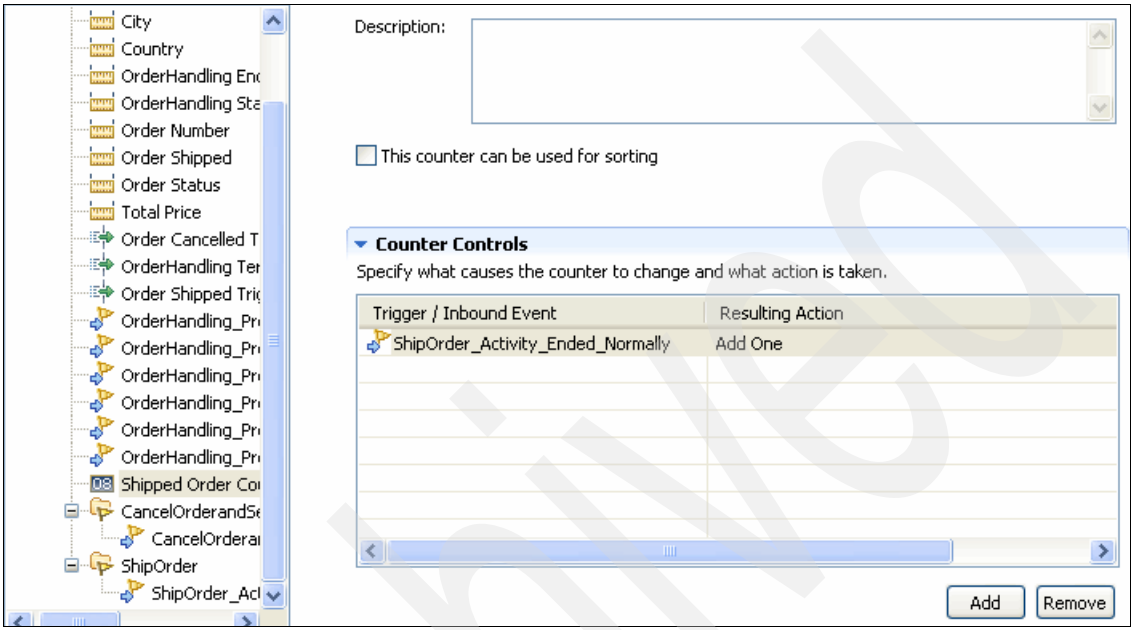


Figure 8-78 Verifying that the Resulting Action is 'Add One'

25. Save your work.

Archived

26. Right-click **OrderHandling** and select **New** → **Stopwatch** (Figure 8-79) to add a stopwatch.

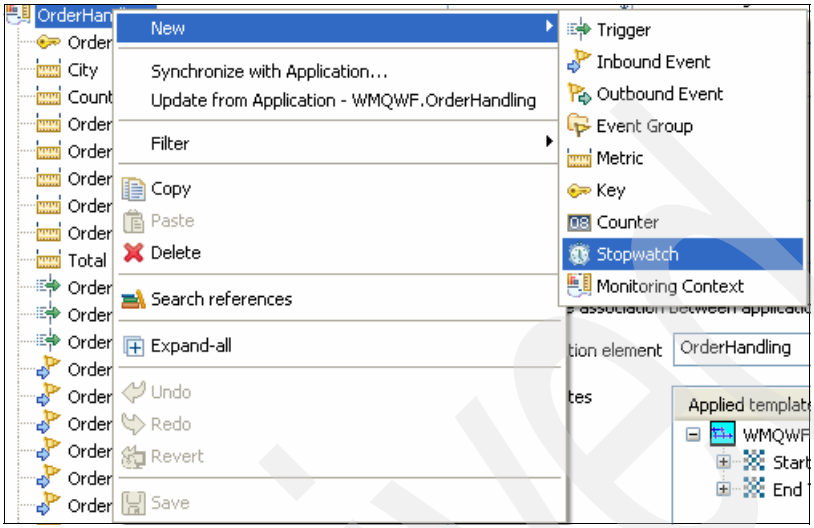


Figure 8-79 Selecting the options to add a stopwatch

27. In the Create New Stopwatch window (Figure 8-80), for Name, type `Order Fulfillment Duration` and click **OK**.

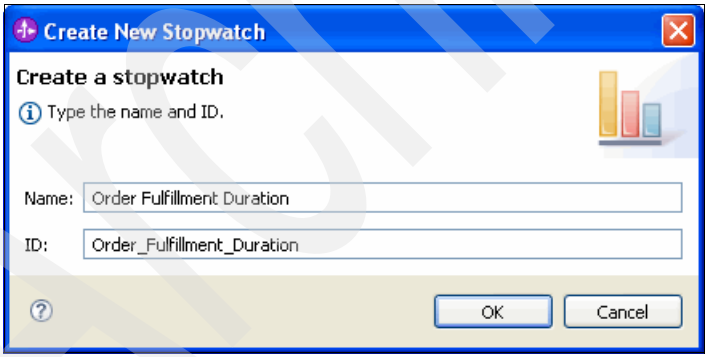


Figure 8-80 Setting the stopwatch name

Archived

28. For the newly created stopwatch, under Stopwatch Controls (Figure 8-81), click the **Add** button.

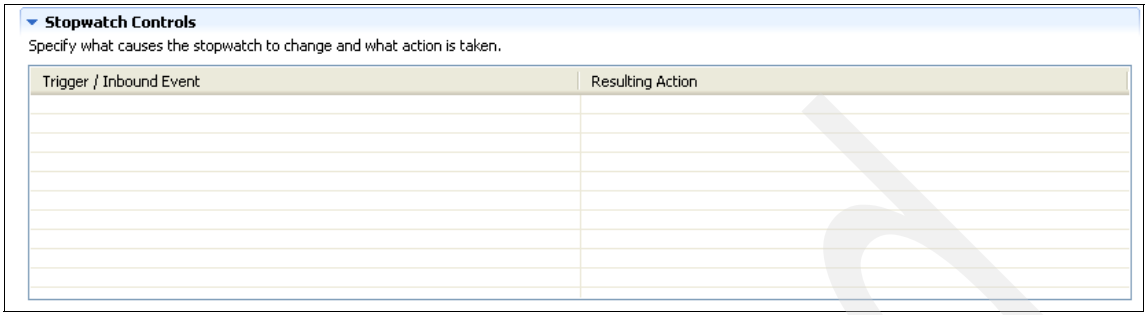


Figure 8-81 Stopwatch Controls area for a stopwatch

29. In the Select Trigger or Inbound Event window (Figure 8-82), click **OrderHandling_Process_Created** and click **OK**.

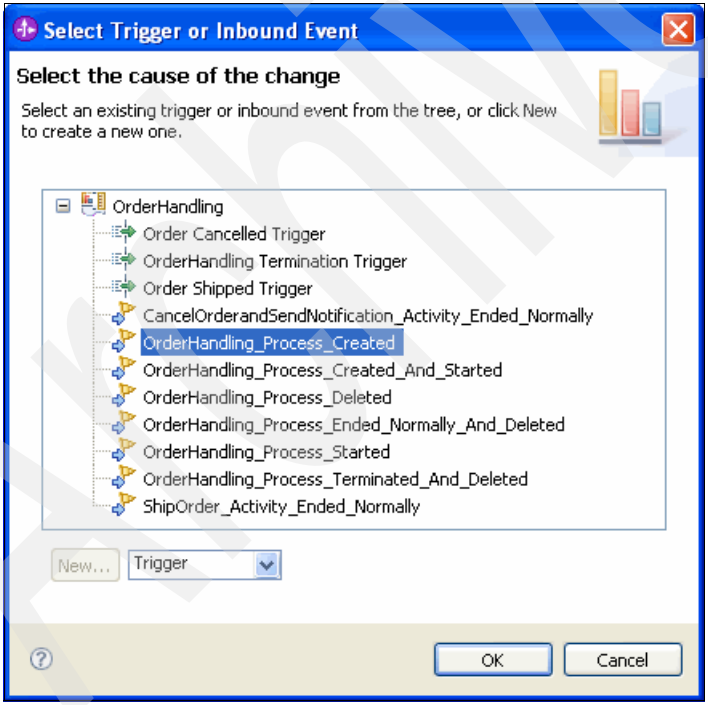


Figure 8-82 Selecting the inbound event for Order Fulfillment Duration

Archived

30. In a similar way, add the other three inbound events and choose the appropriate Resulting Action as follows (Figure 8-83):
    a. For OrderHandling_Process_Created_And_Started, select **Start**.
    b. For CancelOrderandSendNotification_Activity_Ended_Normally, select **Stop**.
    c. For ShipOrder_Activity_Ended_Normally, select **Stop**.

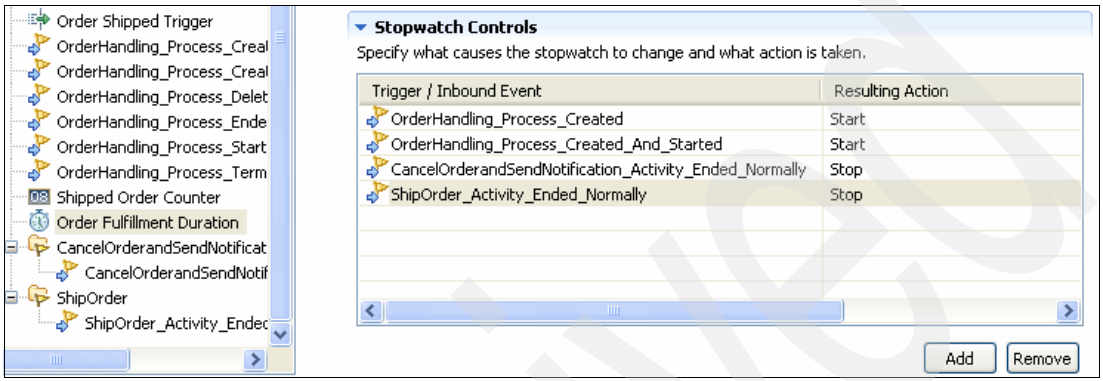


Figure 8-83 Adding inbound events and setting the resulting actions

31. Save your work.
32. Add an outbound event in the monitor model so that when one order handling fulfillment duration is too long, the alert can be sent. Create a new event definition. In the Project Explorer view (Figure 8-84), expand **OrderHandlingBM**, right-click **Event Definitions**, and select **New** → **Event Definition... (cbe)**.

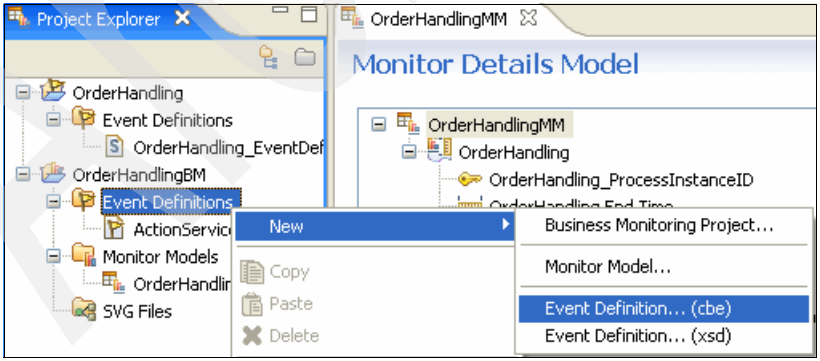


Figure 8-84 Creating a new Common Base Event event definition

33. In the New Event Definition window (Figure 8-85), for File name, type `LateOrderShippedEvent` and click **Finish**.

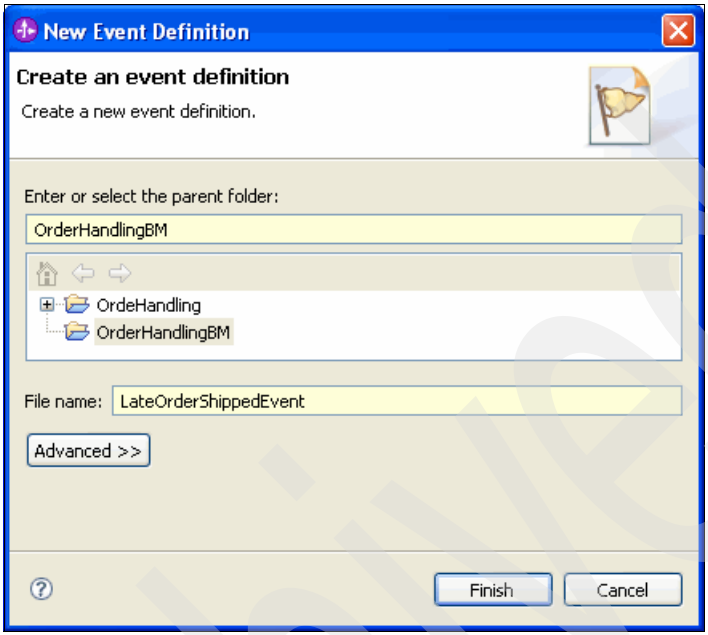


Figure 8-85 Setting the new Common Base Event file name

34. An initial event definition is created. Click the **Browse Parent Event** icon to the right of Parent (Figure 8-86).

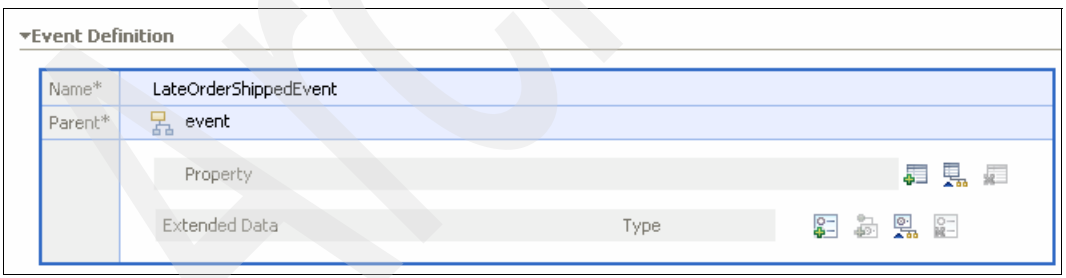


Figure 8-86 The initial event definition

35. In the Select Event Definition window (Figure 8-87), under Matching Events, ensure that **ActionServicesEvent** is selected and then click **Finish**.

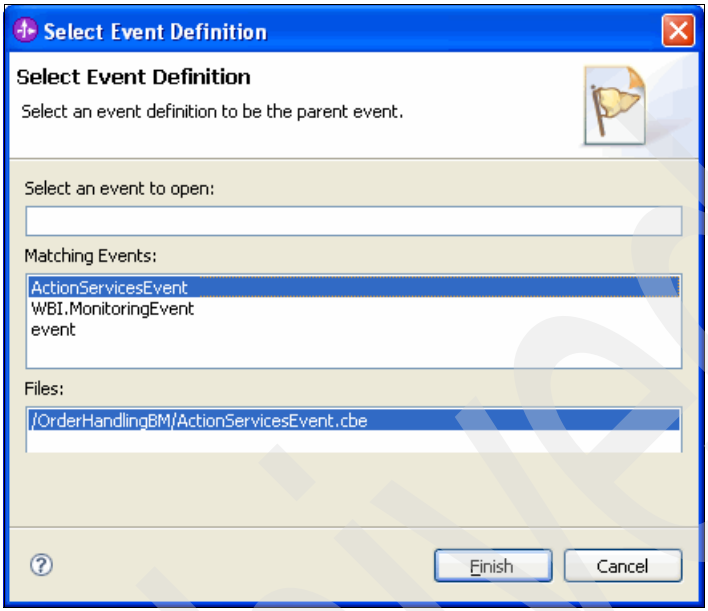


Figure 8-87 Selecting ActionServicesEvent as the parent event

36. Under Event Definition (Figure 8-88), add the extended data:

    a. Click the **Add Extended Data** icon to the right of Extended Data.
    b. For Name, type `OrderProcessingTime` and type to **string**.
    c. Click the **Add Extended Data** icon again.
    d. Change Name to `OrderNumber` and type to **string**.
    e. Save the changes.

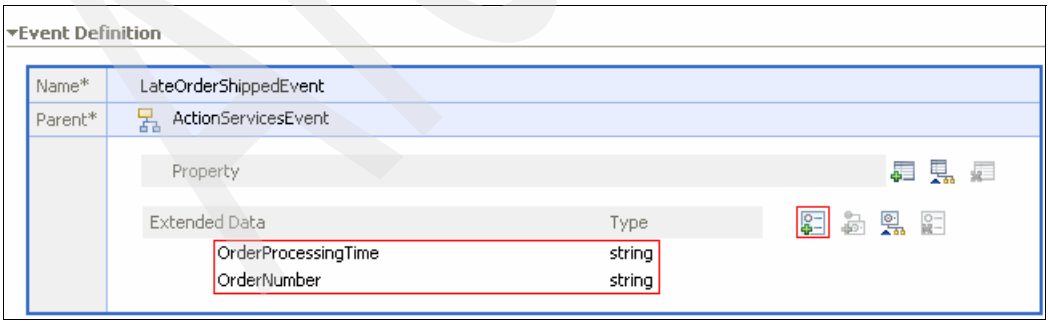


Figure 8-88 Adding the extended data for the event definition

37. Return to the OrderHandlingMM view (Figure 8-89) and add a new trigger:
    a. Name the trigger `Late Order Fulfillment Event`.
    b. Clear the **Trigger is repeatable** option.
    c. Add a Source Type of Recurring wait time and set the interval to 1 minute.
    d. Under Trigger Condition, type `Order_Fulfillment_Duration ge xs:dayTimeDuration('P0DT0H3M0.000S')` to indicate that, when the order is handled over 3 minutes, an alert is sent.

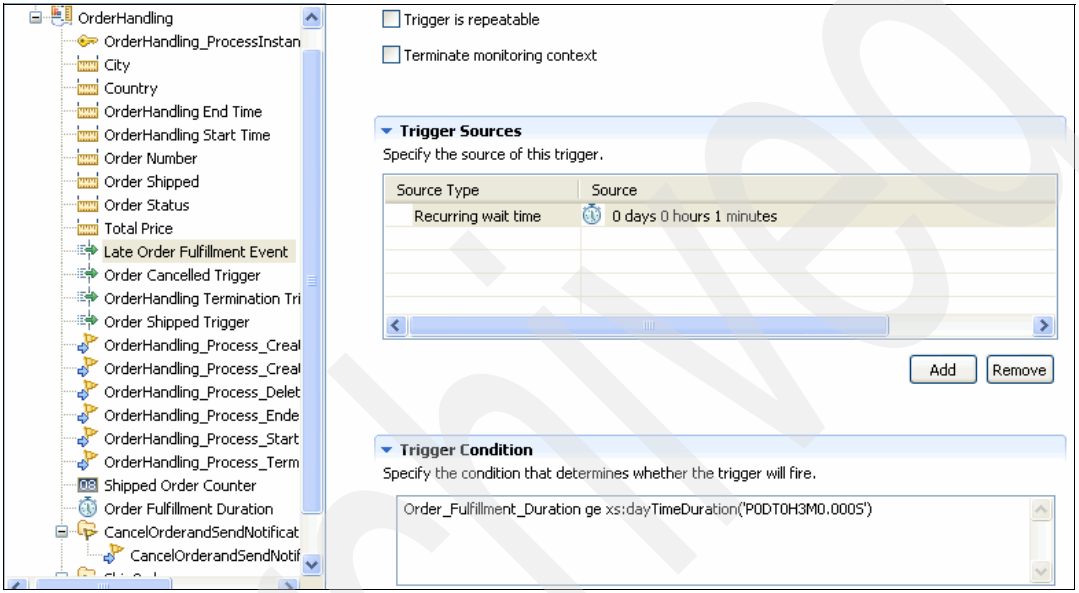


Figure 8-89 Adding the Late Order Fulfillment Event trigger

Archived

38. Right-click **OrderHandling** and select **New** → **Outbound Event** (Figure 8-90) to add an outbound event.

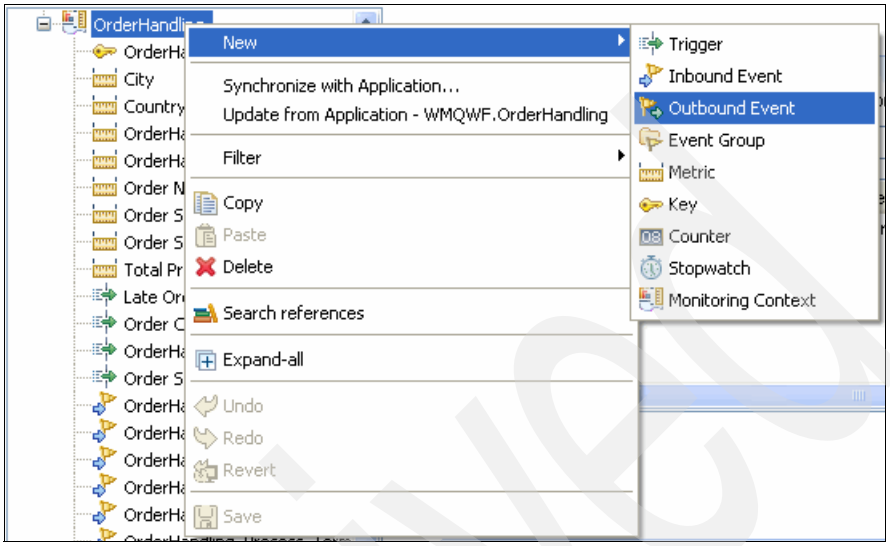


Figure 8-90 Selecting the options to add an outbound event

39. In the Create New Outbound Event window (Figure 8-91):

   a. For Name, type `Late Order Shipped Event`.

   b. Select **Configure this event to be processed by WebSphere Business Monitor action services**.

   c. For Trigger, click **Browse...**.

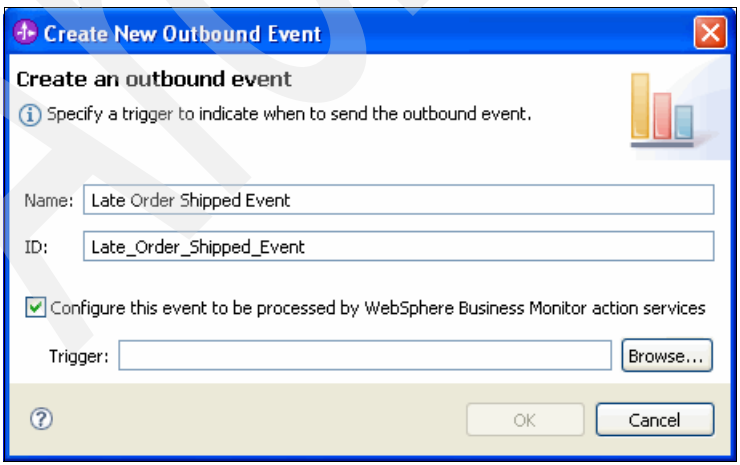


Figure 8-91 Setting the outbound event name

d. In the Select Trigger window (Figure 8-92), select **Late Order Fulfillment Event** and click **OK**.

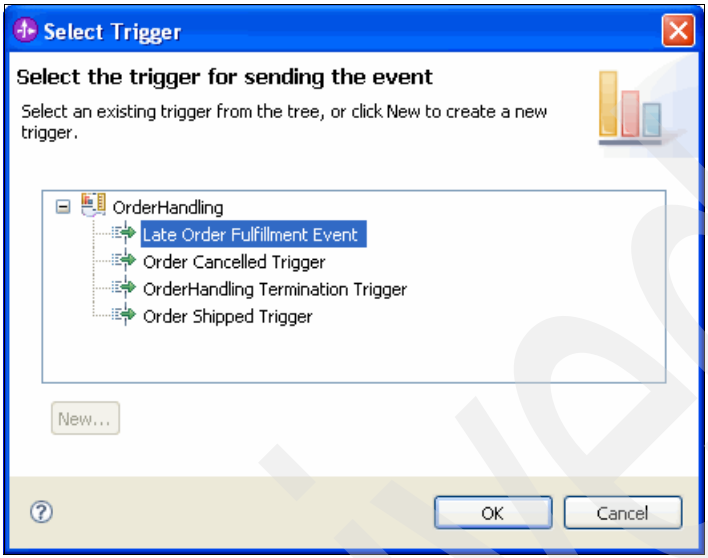


Figure 8-92 Selecting the trigger for the outbound event

e. Back in the Create New Outbound Event window, click **OK**.

40. Under Event Type Details (Figure 8-93), for Extension name, click **Browse...** and select **LateOrderShippedEvent**. Click **OK**.

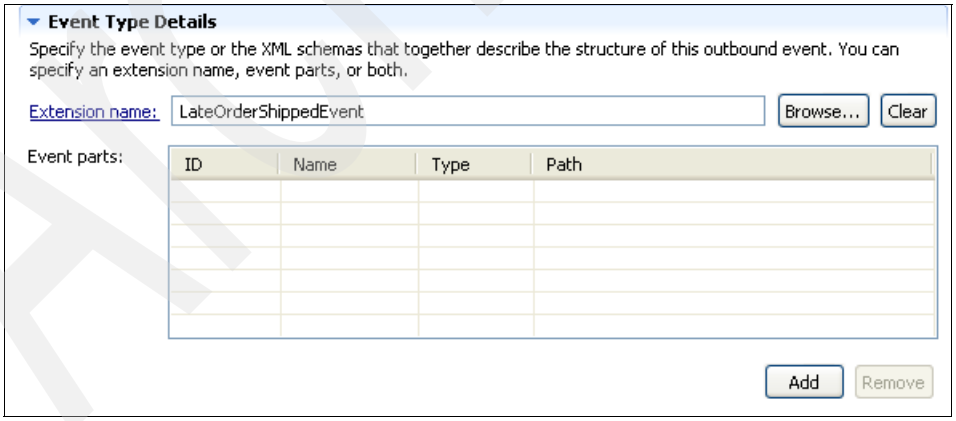


Figure 8-93 Selecting the event type for the outbound event

Archived

41. Under Event Attributes Details (Figure 8-94), set the event attributes details:
    a. Expand **Late Order Shipped Event**.
    b. For the OrderProcessingTime expression, type `xs:string(Order_Fulfillment_Duration)`.
    c. For the OrderNumber expression, type `xs:string(Order_Number)`.
    d. For the BusinessSituationName expression, type `'Order handling duration too long'`.

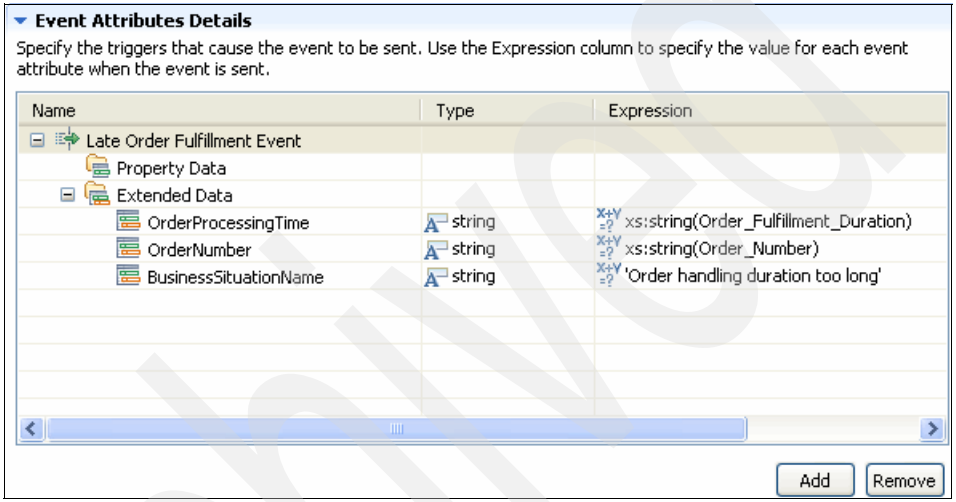


Figure 8-94 Setting the outbound event attributes details

42. Save your work.

Archived

Defining the KPI model

To define the KPI model:

1. Click the **KPI Model** tab on the bottom of OrderHandlingMM view to switch to KPI Model page.
2. In the KPI Model panel (Figure 8-95), right-click **OrderHandlingMM** and then click **New** → **KPI Context** to add a KPI context.

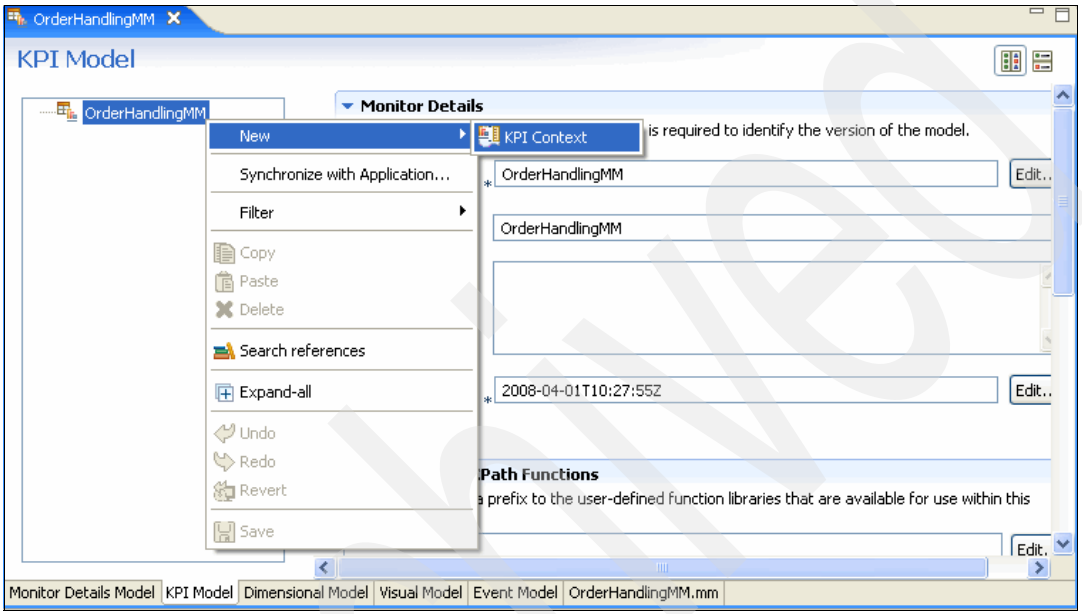


Figure 8-95 Adding a KPI context

3. In the Create New KPI Context window (Figure 8-96), for Name, type `OrderHandlingKC` and click **OK**.

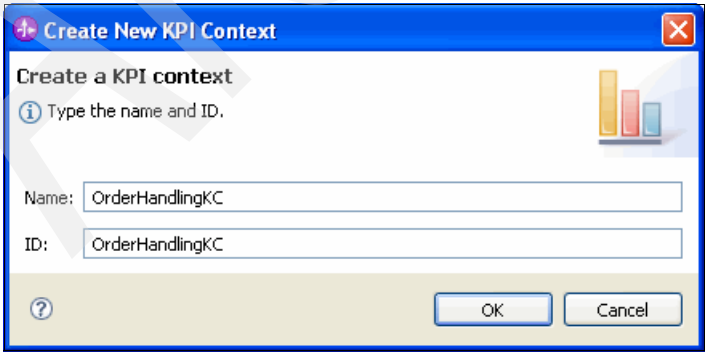


Figure 8-96 Setting the KPI context name

4. Right-click **OrderHandlingKC** and select **New** → **KPI** (Figure 8-97) to add a KPI.

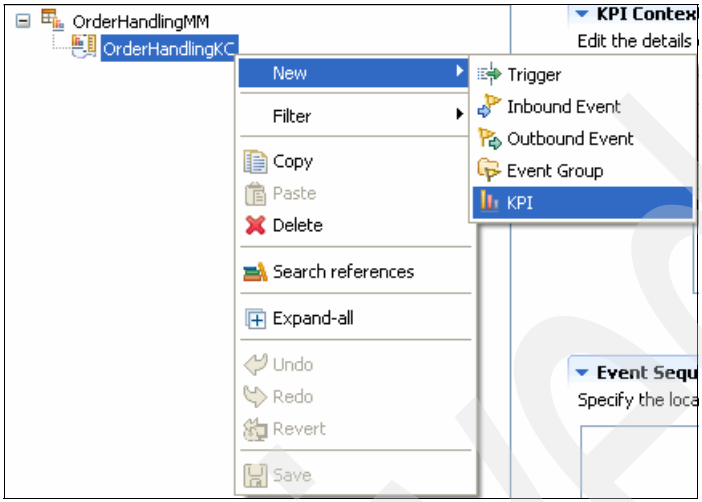


Figure 8-97 Adding a KPI

5. In the Create New KPI window (Figure 8-98), for Name, type `Average Order Fulfillment` and click **OK**. This KPI calculates the average time for the order fulfillment.

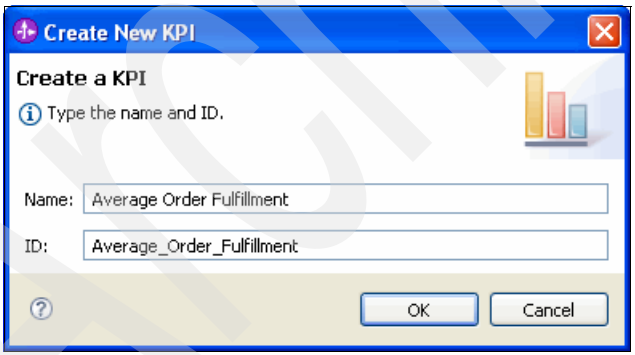


Figure 8-98 Setting the KPI name

6. For the newly created KPI, change Type to **Duration** (Figure 8-99).

Figure 8-99 Changing Type to Duration

7. Under KPI Target and Ranges (Figure 8-100):
   a. Click the **Details** button.

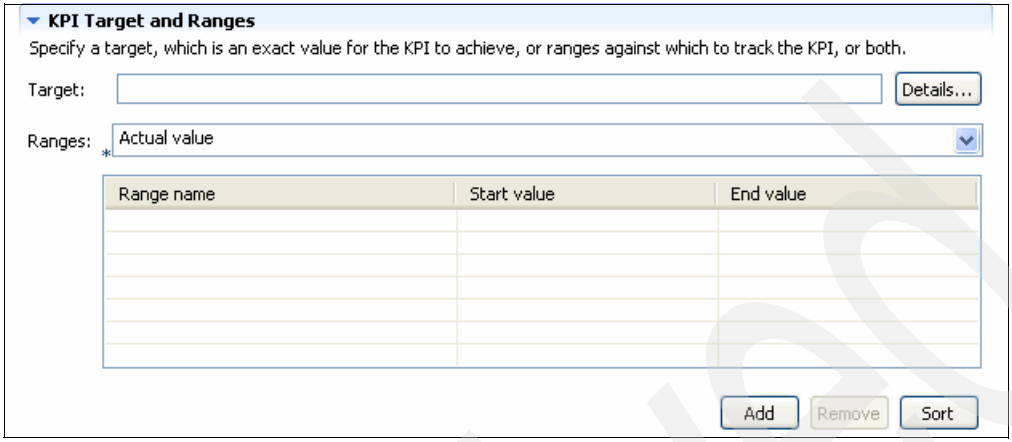


Figure 8-100 KPI Target and Ranges area for a KPI

   b. In the Target Details window (Figure 8-101), for Minutes, select **4**. From a business perspective, it is better to set the target value at the day level. We set the target value and later range values at the minute level so that we can check the monitoring result within a few minutes. Click **OK**.

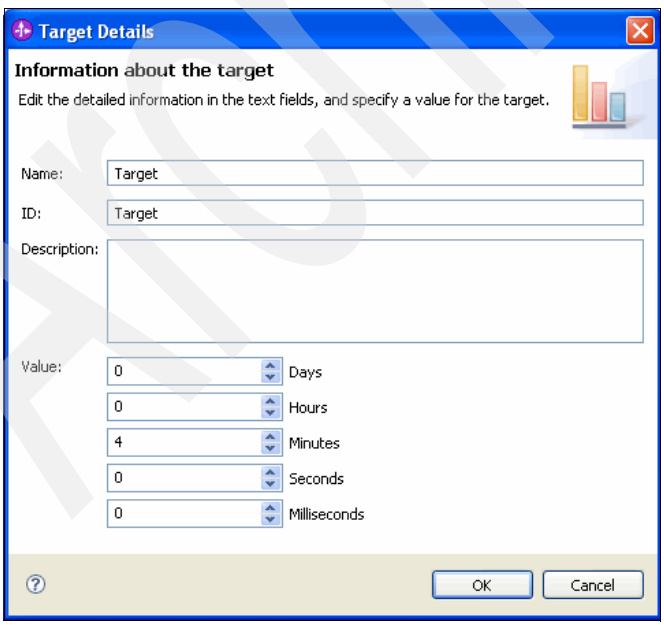


Figure 8-101 Setting the target value

Archive

c. Under KPI Target and Ranges, click the **Add** button.

d. In the Add Range window (Figure 8-102), for Name, type `Day 1-2`. Although we set the target value and range start/end value at the minute level, we still name the ranges at day level from the business point of view. Click **OK**.

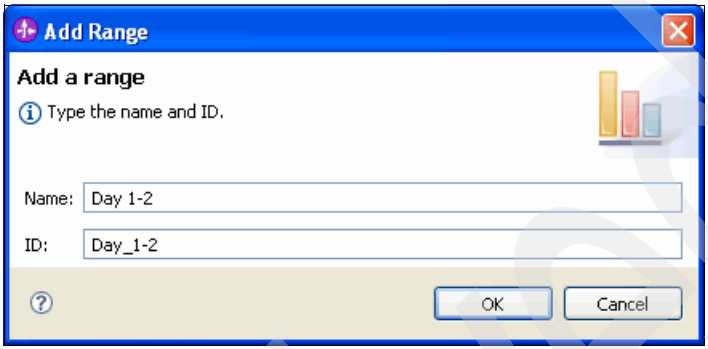


Figure 8-102 Setting the range name

e. For the newly added range, click the button in the End value column and type 2 minutes (Figure 8-103).

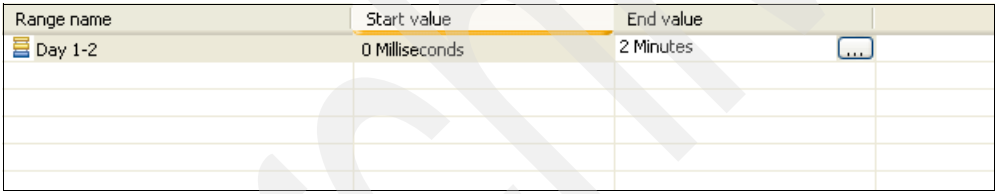


Figure 8-103 Setting the range end value

f. Add the other ranges as follows (Figure 8-104):

i. For Day 3, set the range as 2 Minutes ~ 3 Minutes.
ii. For Day 4-5, set the range as 3 Minutes ~ 10 Minutes.

Range name	Start value	End value
Day 1-2	0 Milliseconds	< 2 Minutes
Day 3	2 Minutes	< 3 Minutes
Day 4-5	3 Minutes	< 10 Minutes

Figure 8-104 All ranges for the KPI

8. Under KPI Value (Figure 8-105), select **Base this KPI on a metric and an aggregation function**.

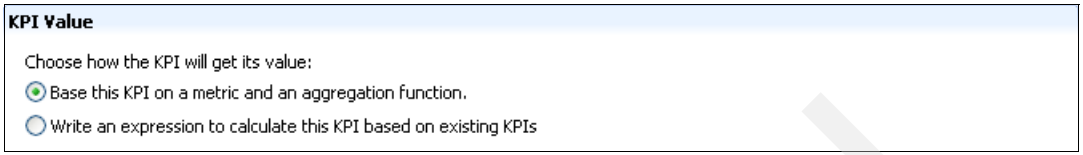


Figure 8-105 Setting the KPI value to be based on an aggregate metric

9. Under KPI Definition or KPI Details:
   a. For Monitoring context, click **Browse...**.
   b. In the Select Monitoring Context window (Figure 8-106), select **OrderHandling** and click **OK**.

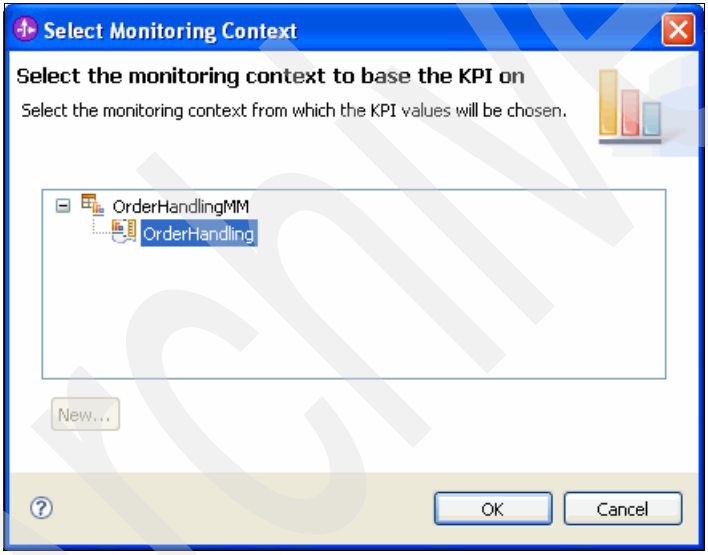


Figure 8-106 Select monitoring context for a KPI

   c. Under KPI Definition or KPI Details, for Metric, click **Browse...**.

Archived

d. In the Select Metric window (Figure 8-107), select **Order Fulfillment Duration** and click **OK**.

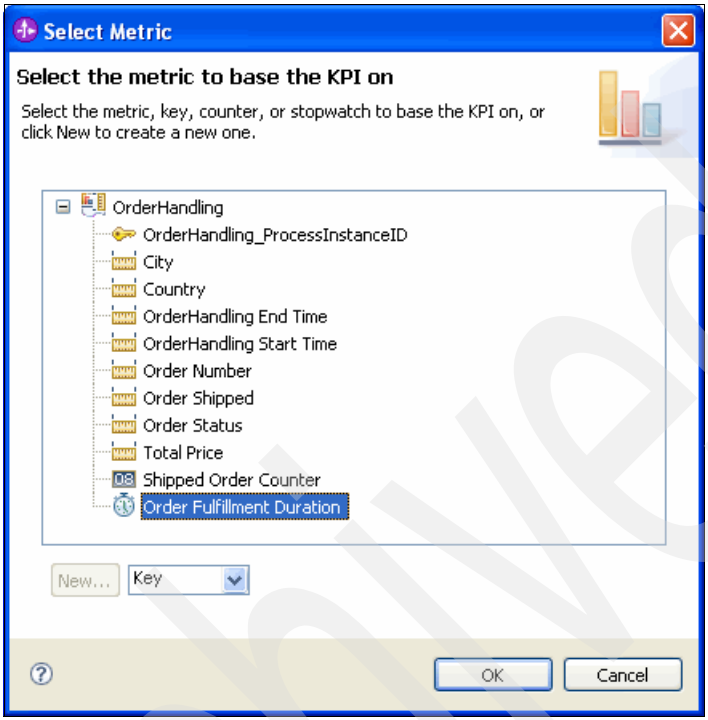


Figure 8-107 Selecting the metric for a KPI

e. Under KPI Definition or KPI Details (Figure 8-108), for Aggregation function, ensure that **Average** is selected.

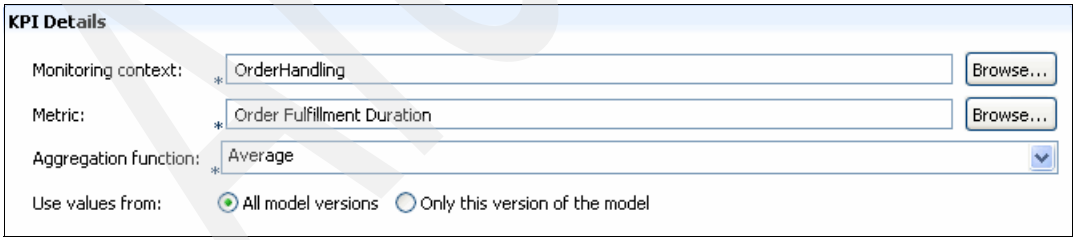


Figure 8-108 Verifying the KPI source and aggregation function

10. Add another KPI with the Name `Percentage of Shipped Orders` and Type `Decimal` (Figure 8-109). This KPI calculates the percentage of shipped orders to total orders.

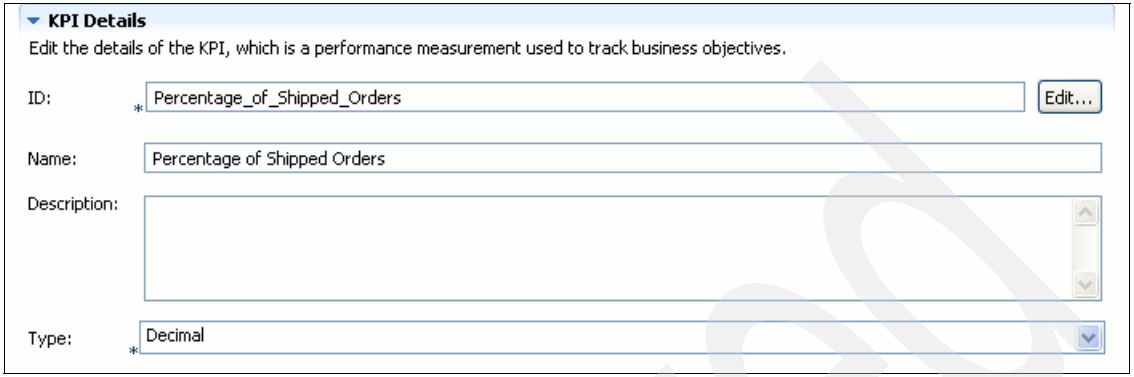


Figure 8-109 Percentage of Shipped Orders KPI basic information

11. Under KPI Target and Ranges (Figure 8-110):

    a. For Target, type `90`.
    b. For Ranges, type.
    c. Set the Low Range to `0~90`.
    d. Set the High Range to `90~100`.

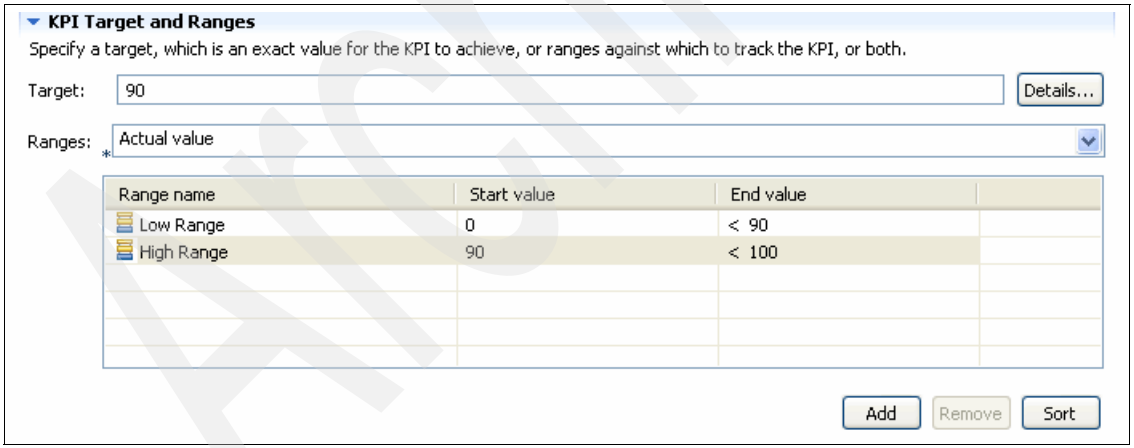


Figure 8-110 Percentage of Shipped Orders KPI target and ranges

Archived

12. Under KPI Details (Figure 8-111), for Metric, select **Order Shipped** for the average value for the Percentage of Shipped Orders KPI definition.

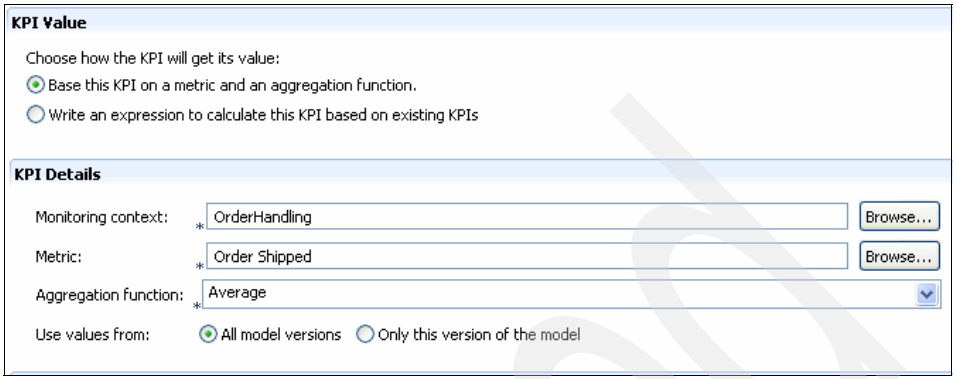


Figure 8-111 Percentage of Shipped Orders KPI definition

13. Save your work.

Defining the dimensional model

To define the dimensional model:

1. Click the **Dimensional Model** tab in the OrderHandlingMM view.
2. In the Dimensional Model panel (Figure 8-112), right-click **OrderHandling Cube** and select **New** → **Dimension** to add a dimension.

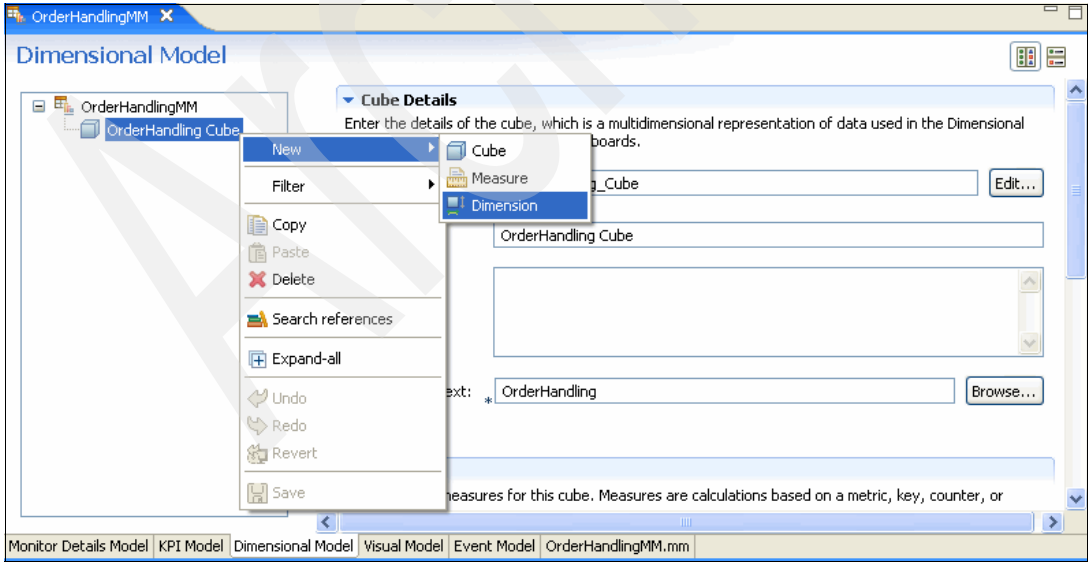


Figure 8-112 Adding a dimension

Archived

3. In the Create New Dimension window (Figure 8-113), for Name, type `Location` and click **OK**.

   This dimension will contain two dimension levels: Country and City. With this dimension, we can drill down or up in the dimensional view on the dashboard to see the statistic data, such as the number of shipped orders, in the different locations.

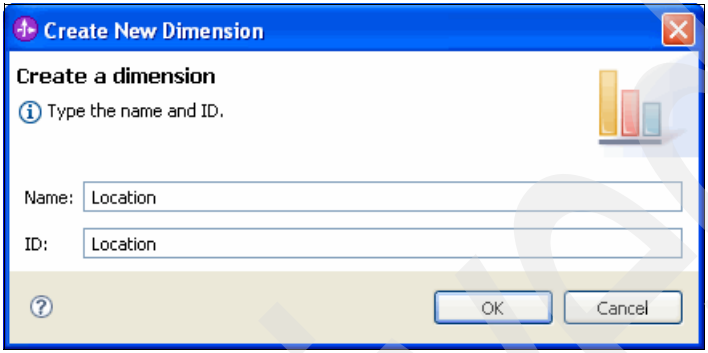


Figure 8-113 Setting the dimension name

4. Right-click **Location** and select **New** → **Dimension Level** to add a dimension level (Figure 8-114).

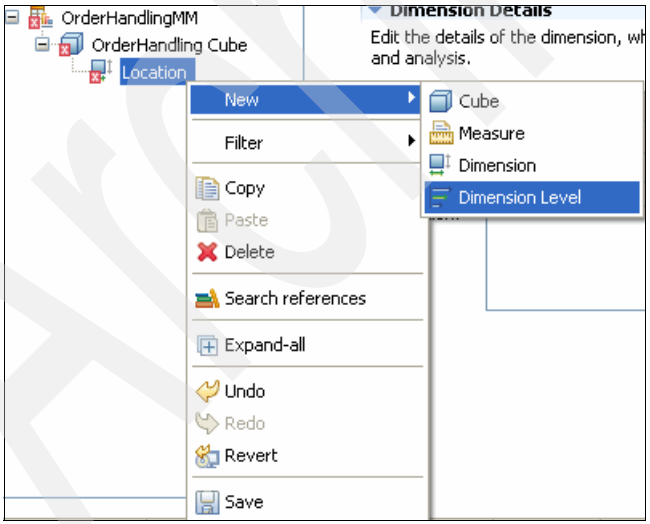


Figure 8-114 Selecting the options to add a dimension level

Archived

5. In the Create New Level window (Figure 8-115):
   a. For Name, type `Country` and click **Browse...**.

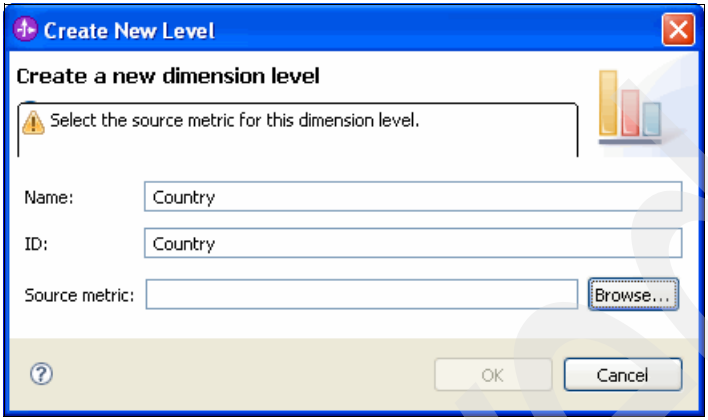


Figure 8-115 Setting the dimension level name

   b. In the Select Metric window (Figure 8-116), click **Country** and click **OK**.

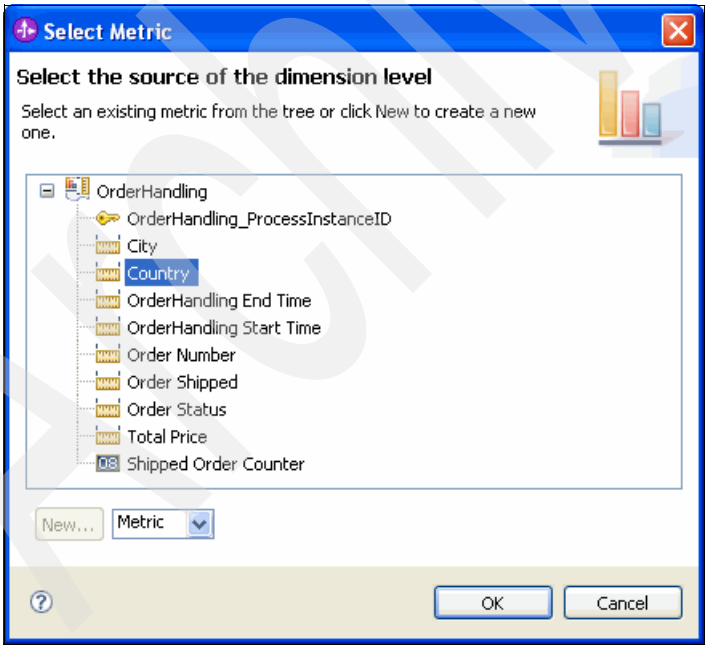


Figure 8-116 Selecting the metric for a dimension level

   c. In the Create New Level window, click **OK**.

d. With the Country dimension level added, ensure that the Level is 1 (Figure 8-117).

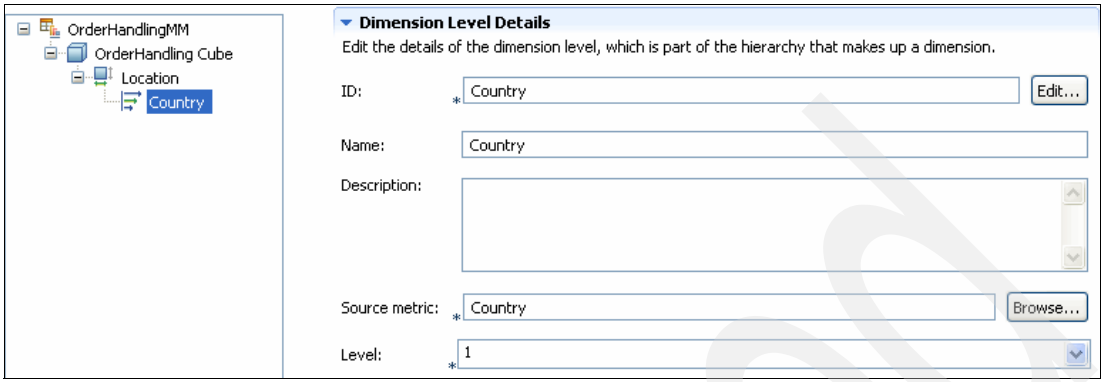


Figure 8-117 Country dimension level

e. In similar way, add a City dimension level as shown in Figure 8-118.

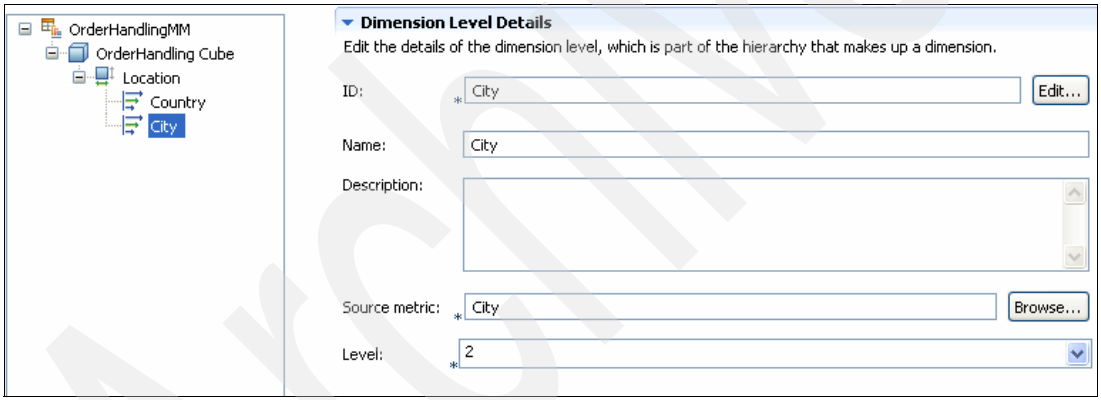


Figure 8-118 City dimension level

Archived

6. Right-click **OrderHandling Cube** and select **New** → **Measure** (Figure 8-119) to add a measure. We define the aggregate measure to calculate the sum of shipped orders.

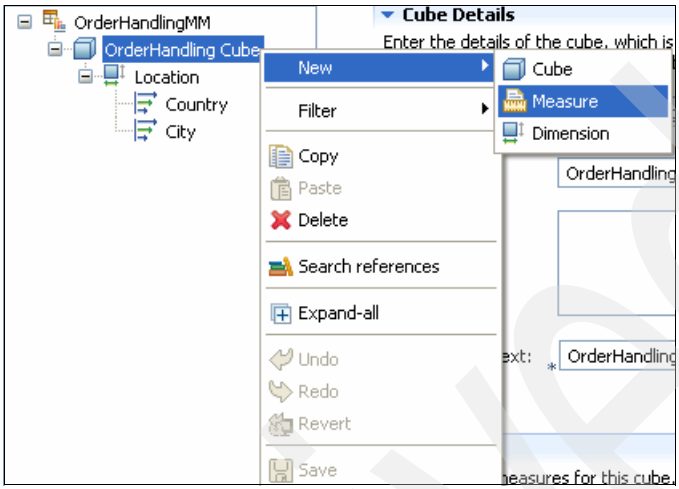


Figure 8-119 Adding a measure

7. In the Create New Measure window (Figure 8-120):
   a. For Name, type `Number of Shipped Orders`.
   b. For Source metric, click **Browse...** and select **Shipped Order Counter**.
   c. For Aggregation function, select **Sum**.
   d. Click **OK**.

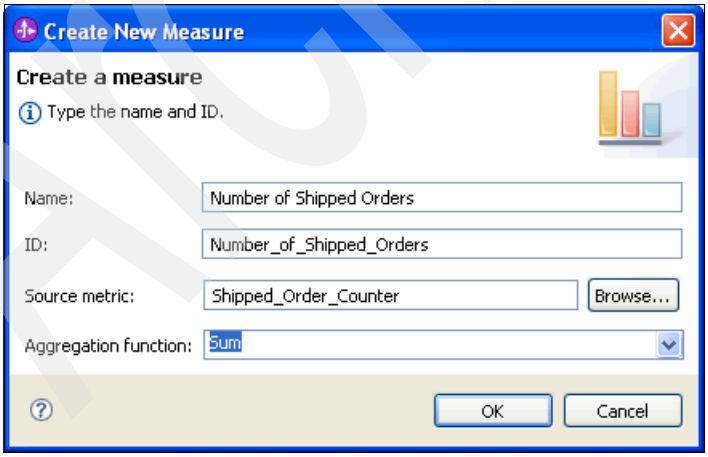


Figure 8-120 Setting the measure name, metric, and aggregation function

8. Save your work.

Archived

Defining the visual model

To define the visual model:

1. Import the following SVG files into our monitor project:
   - Order_Handling_MDM_Order_Handling_MC.svg
   - Order_Handling_KM_Order_Handling_KC.svg

   These files are supplied with the additional materials for this book. To learn more about how to download these files, see Appendix B, “Additional material” on page 801.

   a. In WebSphere Integration Developer, select **File** → **Import...**.

   b. In the Import window (Figure 8-121), expand **General** and select **File System** and click **Next**.

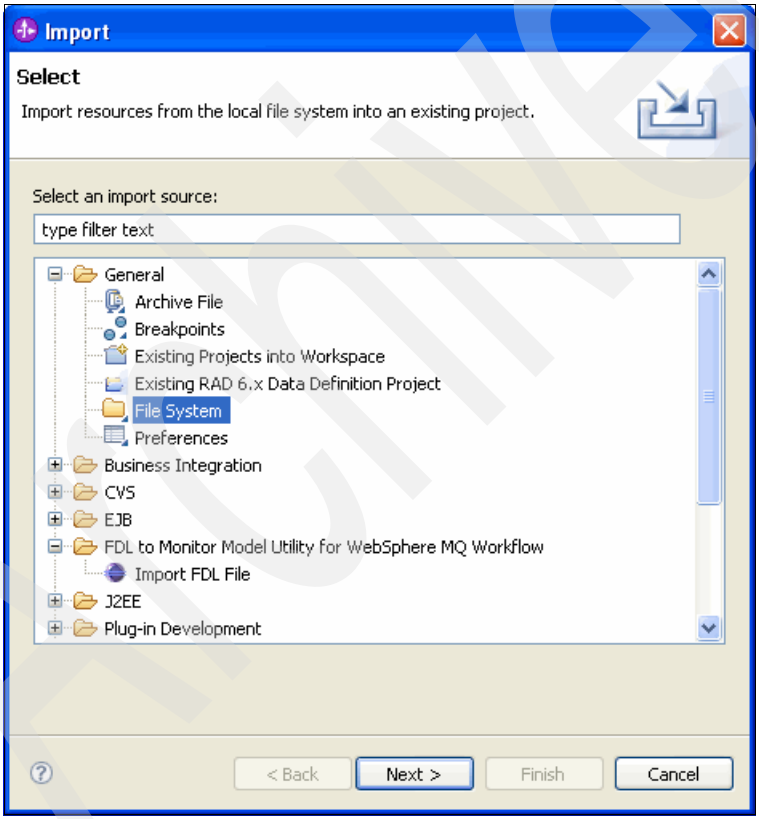


Figure 8-121 Selecting to import from the file system

Archived

c. In the Import – File system window (Figure 8-122):

   i. For From directory, click **Browse**.

   ii. Find the directory where the .svg files are and select the files.

   iii. For the Into folder, make sure the OrderHandlingBM monitor project is selected.

   iv. Click **Finish**.

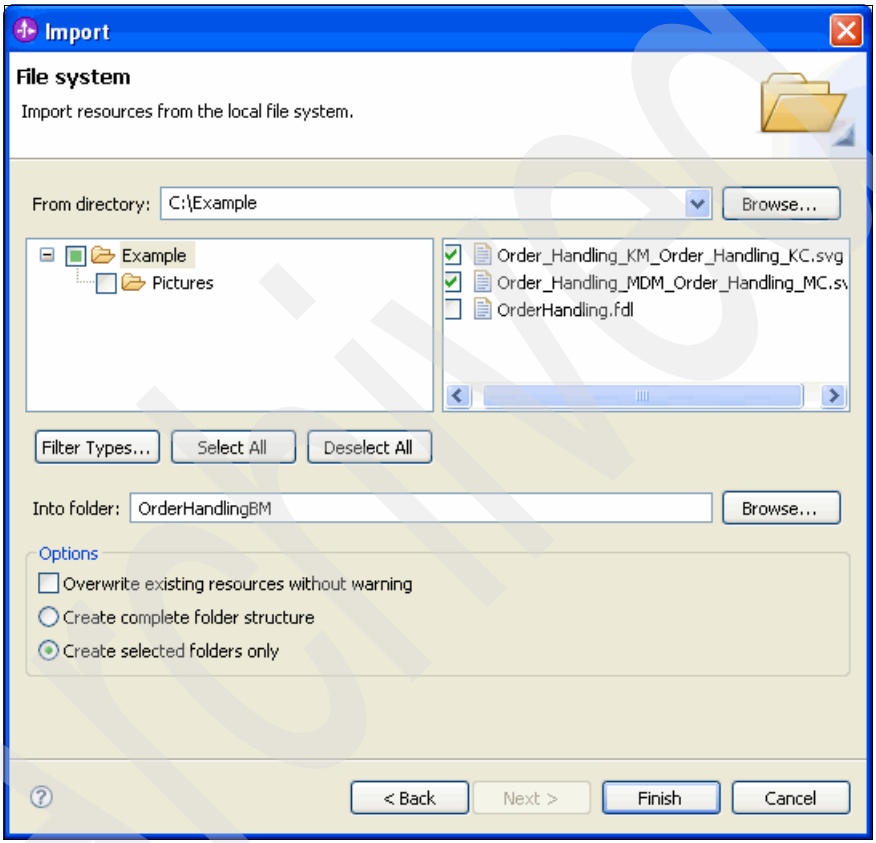


Figure 8-122 Selecting the .svg files to import

2. Click the **Visual Model** tab on the bottom of OrderHandlingMM view. In the Visual Model panel (Figure 8-123):

   a. Click **Browse...**.

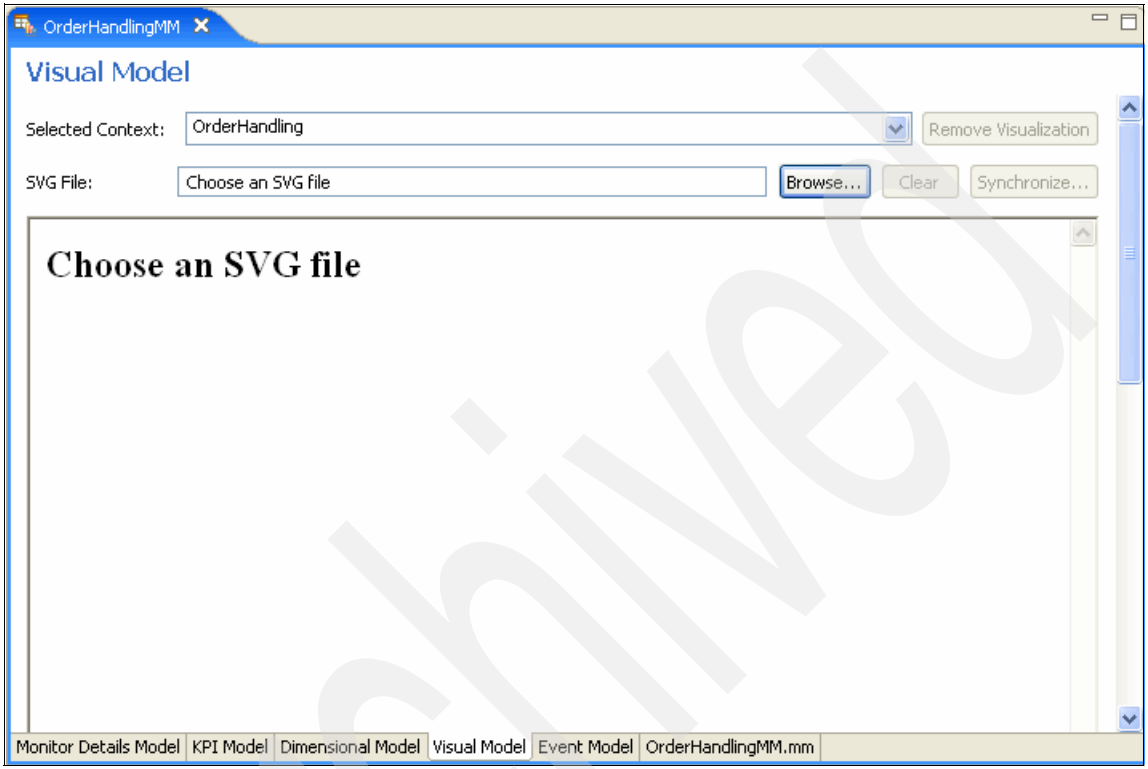


Figure 8-123 Visual model

Archive

b. In the Select SVG Resource window (Figure 8-124), expand **OrderHandlingBM** and select **Order_Handling_MDM_Order_Handling_MC.svg**.

This .svg file is the visualization for the monitoring context defined in the monitor details model and can show each instance status in the diagrams view in the dashboards. In the following steps, we define a set of actions to describe how and when the diagram will be modified based on the values of metrics.

Click **OK**.

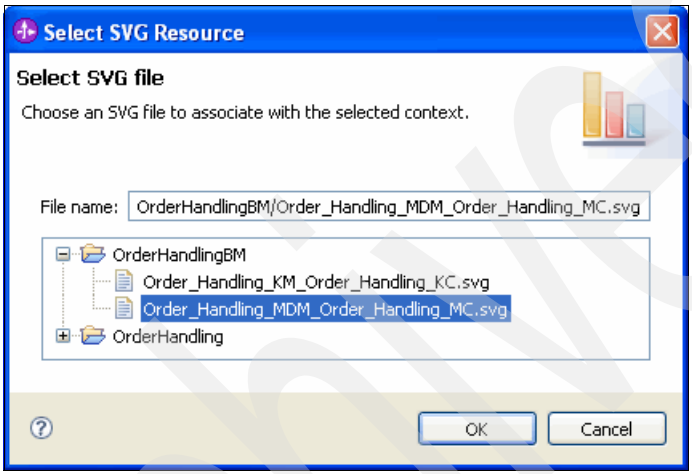


Figure 8-124 Selecting the .svg file for the monitoring context

c. In the Create Shape Sets window (Figure 8-125), click **Yes**.

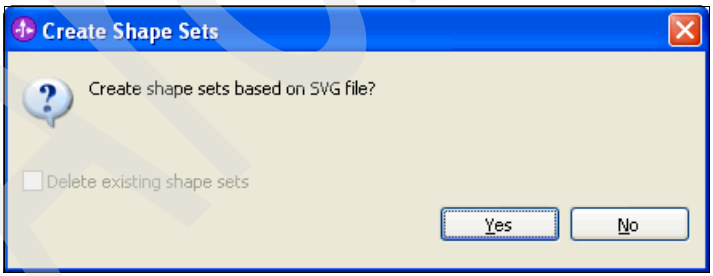


Figure 8-125 Create Shape Sets window

3. Scroll down to Add and Edit Shape Set Actions (Figure 8-126):

   a. Find the **Cancel_Order_and_Send_Notification** shape set and click **Add...**.

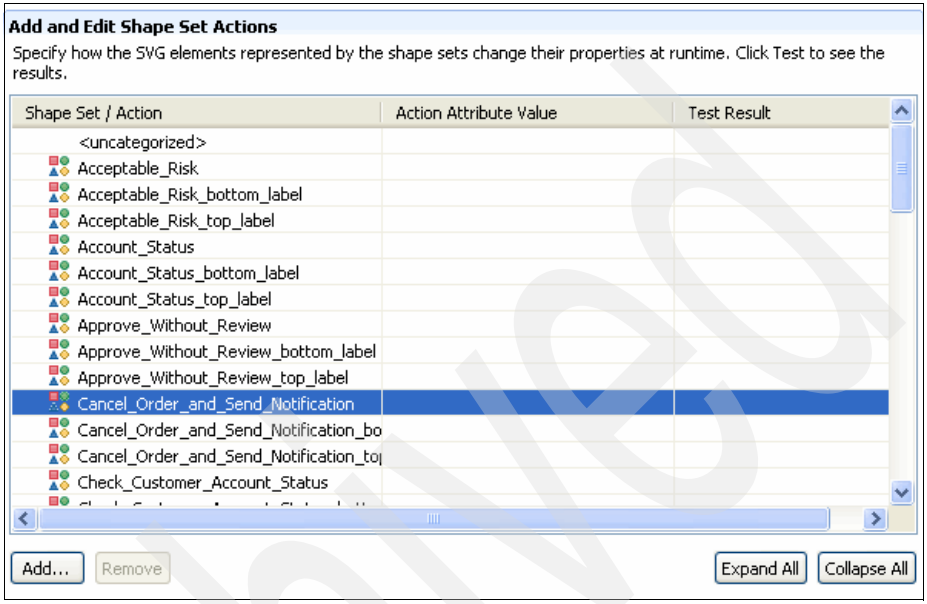


Figure 8-126 Finding the Cancel_Order_and_Send_Notification shape set

   b. In the Add Action window (Figure 8-127), click **Set Color** and click **OK**.

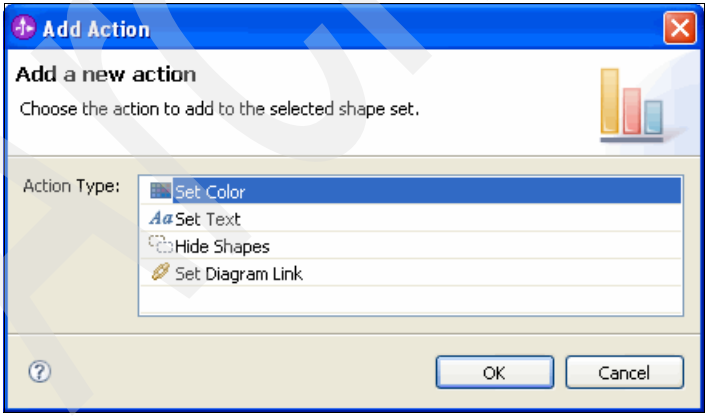


Figure 8-127 Adding the Set Color action

Archived

c. Set the condition and fill color so that, when the condition becomes true, the color is filled (Figure 8-128):

   i. For condition, type `Order_Status eq 'Cancelled'`.
   ii. For fill color, select the **red** color.

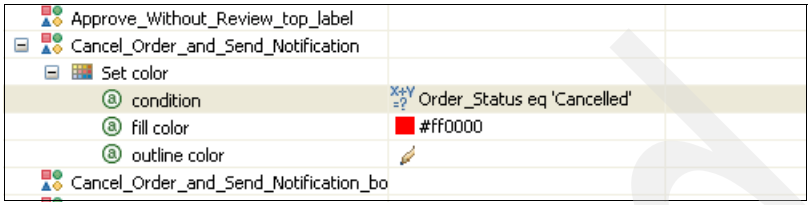


Figure 8-128 Setting the condition and fill color

d. In a similar way, find the Ship_Order shape set and add an action with a condition (`Order_Status eq 'Shipped'`) and fill color (green) (Figure 8-129).

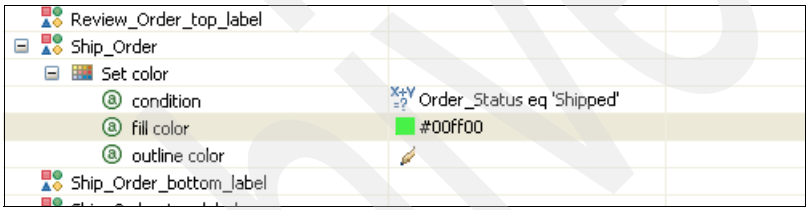


Figure 8-129 Adding the action for Ship_Order shape set

e. Save your work.

4. Scroll up to Selected Context and select **OrderHandlingKC**, and for SVG file, select **/OrderHandlingBM/Order_Handling_KM_Order_Handling_KC.svg** (Figure 8-130).

   This .svg file is the visualization for the KPI context defined in the KPI model and can show KPI status in the diagrams view in the dashboards. In the following steps, we define a set of actions to describe how and when the diagram will be modified based on the values of KPIs.

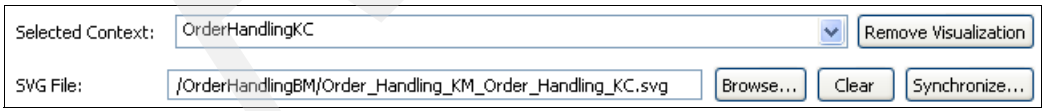


Figure 8-130 Setting the .svg file for KPI context

5. In the Create Shape Sets window, click **Yes**.

Archived

6. Scroll down to Add and Edit Shape Set Actions. Add actions for Ship_Order_2 and Ship_Order_2_bottom_label shape sets (Figure 8-131):

   a. Find the **Ship_Order_2** shape set and add a Set color action. For condition, type `Percentage_of_Shipped_Orders ge 90`, and for fill color, select **green**.

   b. Find the **Ship_Order_2** shape set and add a Set color action. For condition, type `Percentage_of_Shipped_Orders lt 90`, and for fill color, select **red**.

   c. Find the **Ship_Order_2_bottom_label** shape set and add a Set text action. For text value, type `fn:concat('Percentage of shipped orders: ', Percentage_of_Shipped_Orders, '%')`.

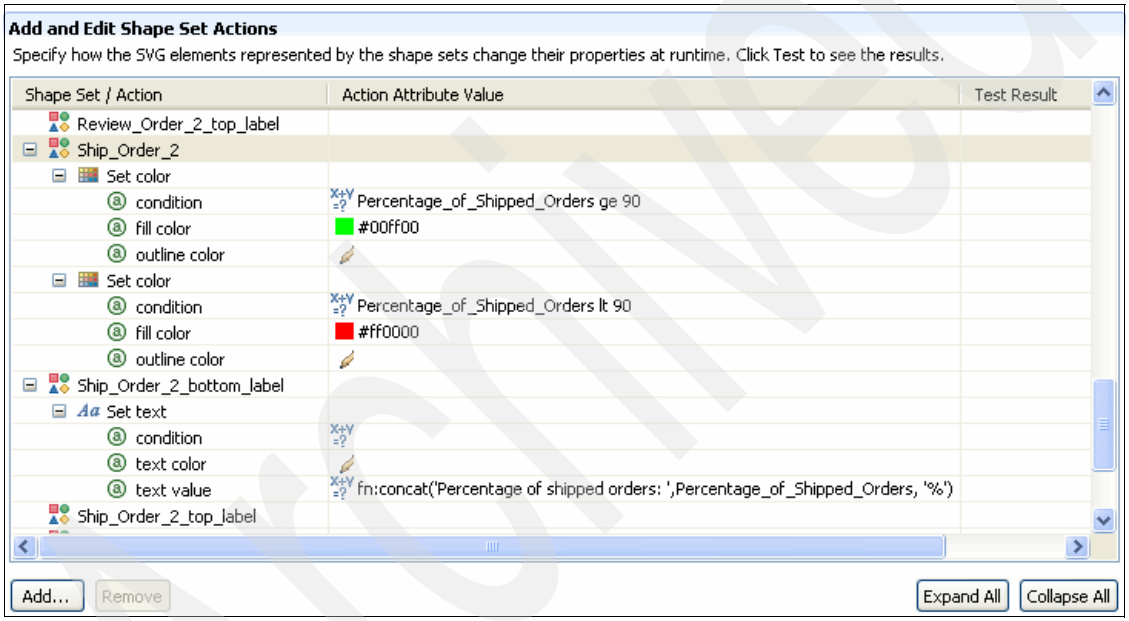


Figure 8-131 Adding actions for KPI context

7. Save your work.

Archive

Generating the J2EE project

In this section, we finish the development of the monitor model by generating the J2EE project:

1. In the Project Explorer view, expand **OrderHandlingBM** → **Monitor Models**, right-click **OrderHandlingMM.mm** and select **Generate Monitor J2EE Projects**.
2. In the Generate Monitor J2EE Projects window (Figure 8-132), accept the default project names and click **Finish**.

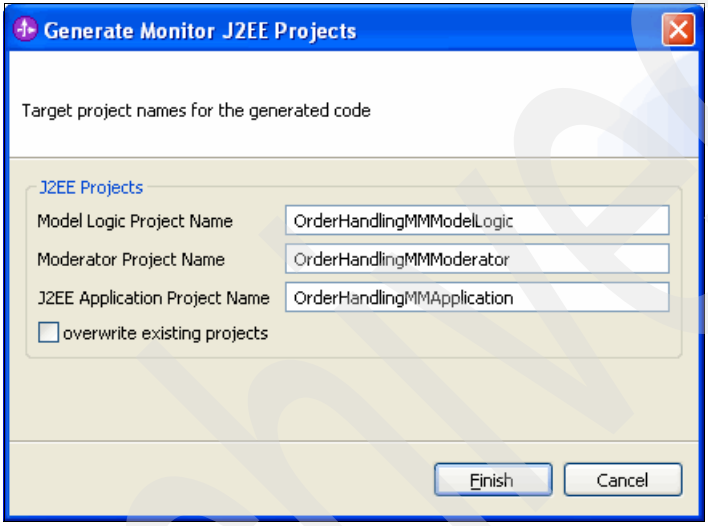


Figure 8-132 Generate Monitor J2EE Projects window

3. After the projects are generated, switch to the J2EE perspective.
4. In the Project Explorer view, right-click **OrderHandlingMMApplication** and select **Export** → **EAR file** to export the EAR file. In this sample, the default file name `OrderHandlingMMApplication.ear` is used.

8.6 Deployment and monitoring

We have now implemented the WebSphere MQ Workflow process and monitor model. Next we complete the following tasks:

- Deploy the WebSphere MQ Workflow process.
- Deploy the monitor model.
- Run instances of the WebSphere MQ Workflow process.
- See the monitoring results on the WebSphere Business Monitor dashboard.

8.6.1 Deploying the WebSphere MQ Workflow process

To deploy the WebSphere MQ Workflow process:

1. On the WebSphere MQ Workflow server, open a Command Prompt by running **cmd.exe**.
2. Navigate to the directory that contains the OrderHandling.fdl file.
3. Import the process into WebSphere MQ Workflow run time by entering the following command (Figure 8-133):

   ```
   fmcibie -uADMIN -ppassword -i"OrderHandling.fdl" -t -x -yFMC
   ```

   Use the `-x` option of the **fmcibie** command to ensure that all names are XML-compliant. Noncompliant names lead to exceptions when parsing the generated Common Base Events.

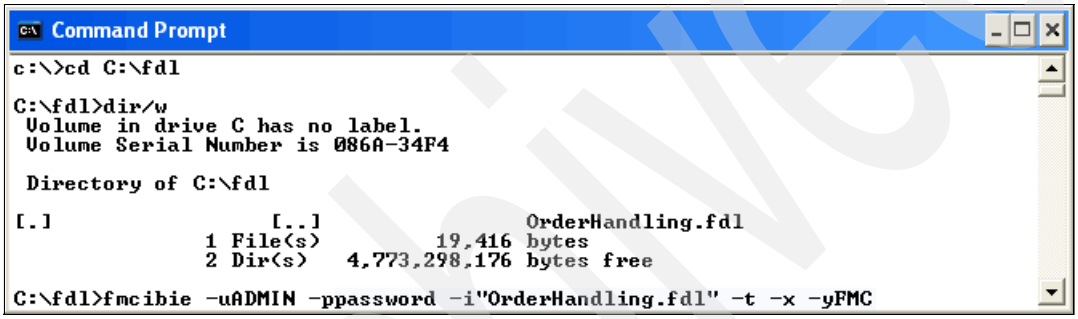


Figure 8-133 Importing the WebSphere MQ Workflow process into run time

Archived

The command completes without error messages as shown in Figure 8-134.

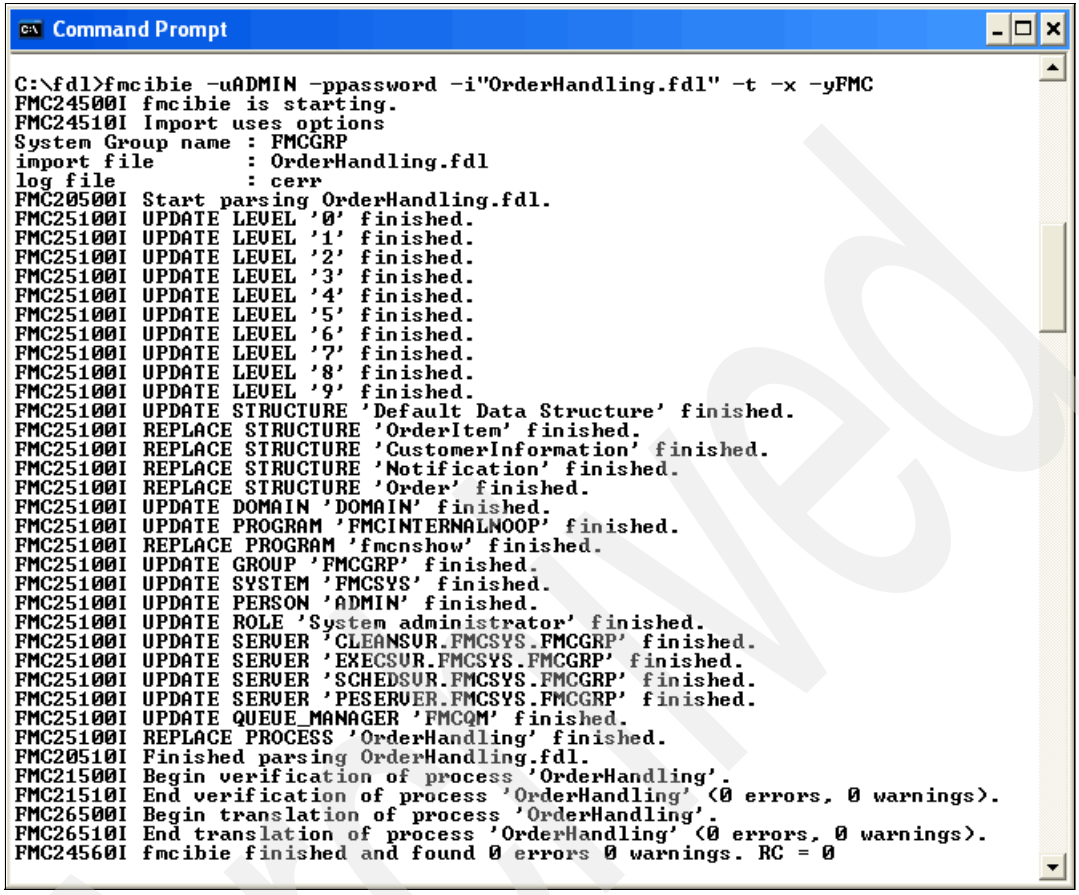


Figure 8-134 WebSphere MQ Workflow process successfully deployed

8.6.2 Deploying the monitor model

To deploy and configure the monitor model:

1. Start the WebSphere Business Monitor server.
2. Open the WebSphere Business Monitor administrative console.

Archived

3. Install the monitor model EAR file. Because in this sample, the CEI server and Alphablox reside in the WebSphere Business Monitor server, we do not need to show all the installation options.

   As shown in Figure 8-135, select **Prompt me only when additional information is required**. For full instructions to install the monitor model, see 3.4.1, “Installing a monitor model by using the default settings” on page 91.

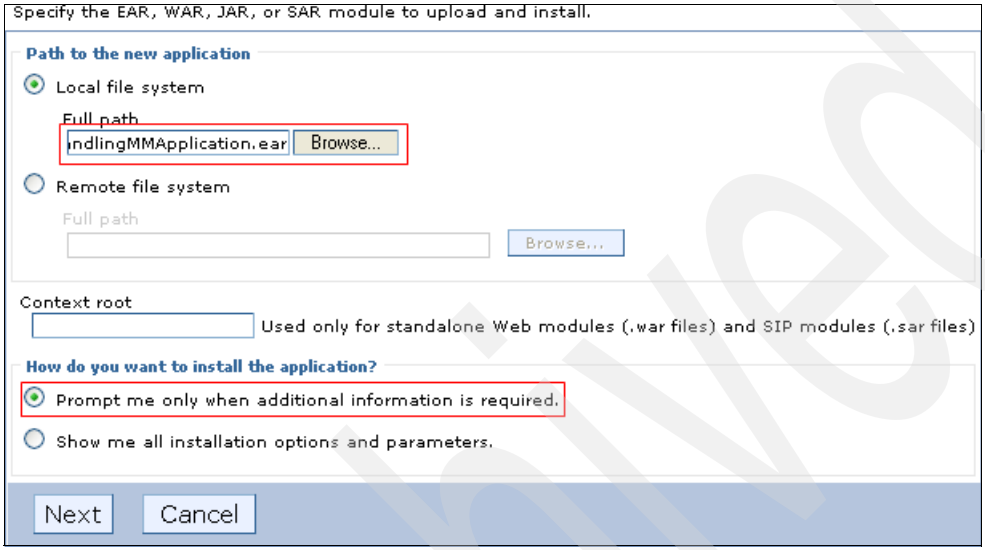


Figure 8-135 Deploying the monitor model EAR file

After monitor model installation completes, the monitor model is started after a while (Figure 8-136).

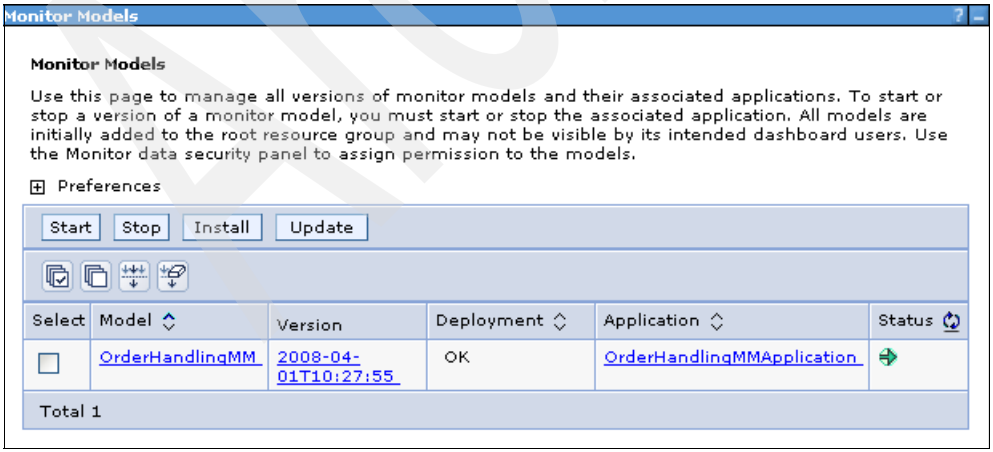


Figure 8-136 Monitor model started

Archived

4. In the left pane of the administrative console (Figure 8-137), expand **Applications** → **Monitor Action Services** → **Template Definitions** and click **Notifications**. In the right pane, click the **New** button.

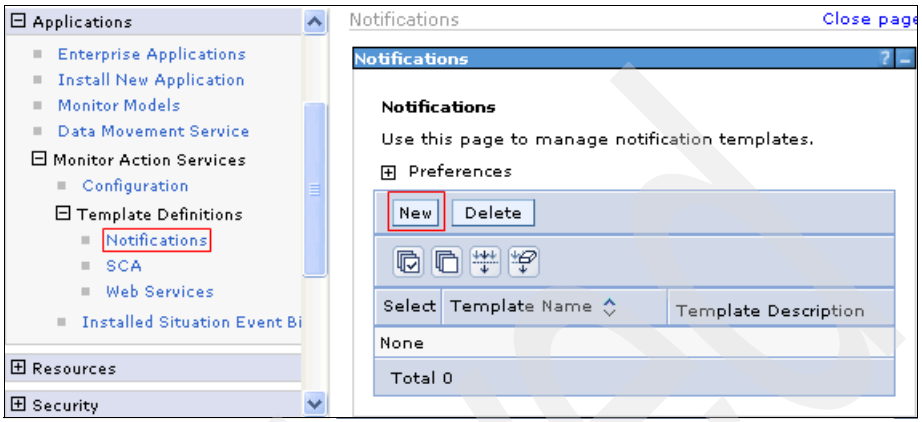


Figure 8-137 Adding a new notification template

5. Under Notification Template Configuration (Figure 8-138 on page 579), define the new notification template to describe the kind of notification that is sent for the specified situation. In this sample, we use a dashboard alert to notify the dashboard user:

   a. For Template name, type `TooLongAlert`.

   b. For Default action service type, select **Dashboard Alert**.

   c. For 'To' query type, select **User id**.

   d. For To, type `admin`.

      Later when we check the dashboard alert, we use `admin` to log in to the dashboard.

   e. For Subject, type `TooLongAlert`.

   f. For Body, type `The order %OrderNumber% is handled over %OrderProcessingTime%`.

   g. Click **OK**.

Archive

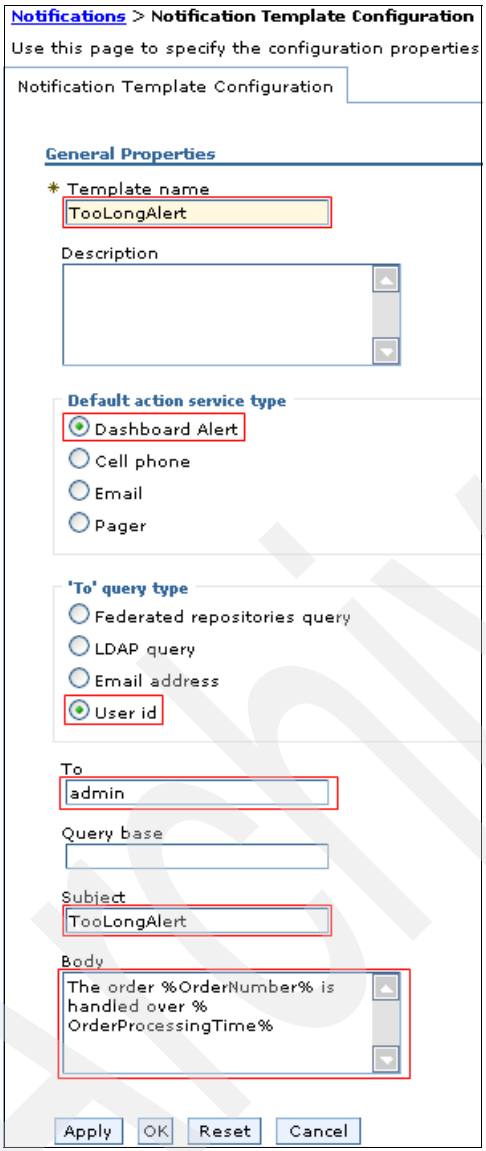


Figure 8-138 Configuring the new notification template

Archived

6. In the left pane of the administrative console (Figure 8-139), expand **Applications** → **Monitor Action Services** and click **Installed Situation Event Bindings**. In the right pane, click the **New** button.

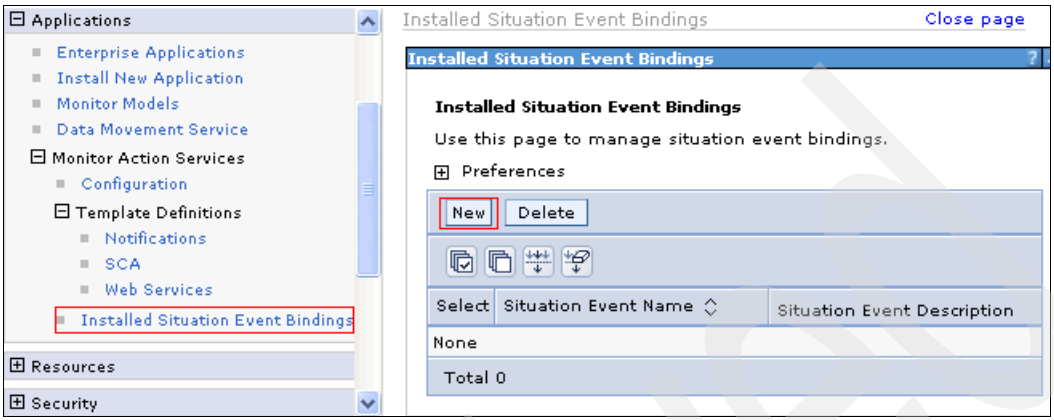


Figure 8-139 Adding a new situation event binding

7. Specify the situation we are interested in and bind it with the notification template defined in previous steps. Thus when the specified situation event comes in, the associated dashboard alert is sent and shown in the dashboard Alerts view.

   On the New Situation Event Binding page (Figure 8-140), for Situation event name, type `Order handling duration too long` and click **Apply**. This name is also set for the attribute BusinessSituationName of the Late Order Shipped Event outbound event.

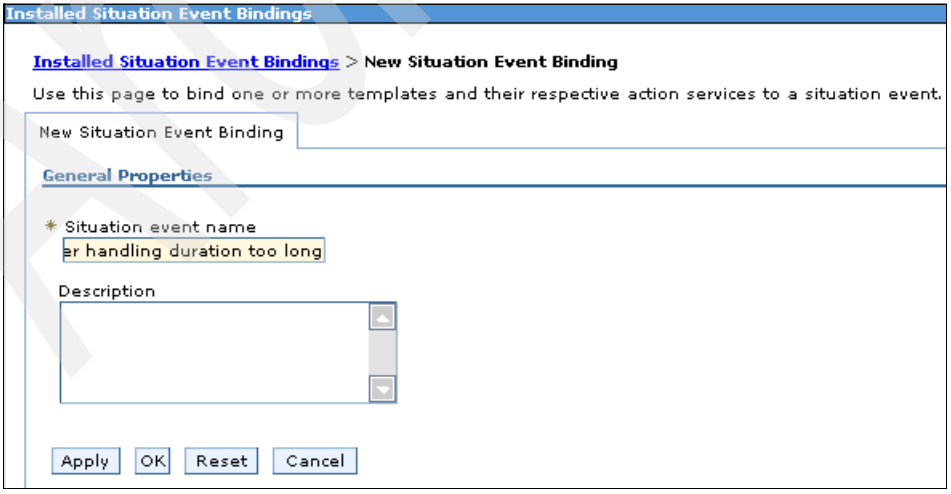


Figure 8-140 Setting the situation event name

8. Click the **Add** button.
9. Under General Properties (Figure 8-141), for Binding name, type `TooLongAlertBinding`, and for Template name, select **TooLongAlert**. Click **OK**.

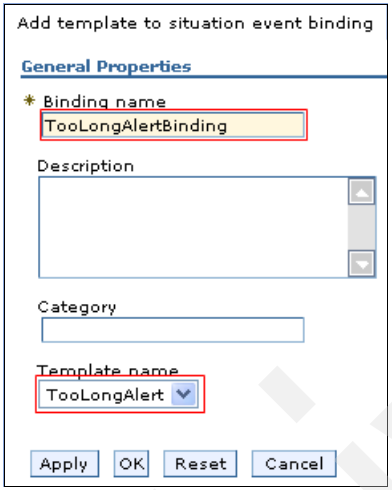


Figure 8-141 Adding the binding

10. Restart the WebSphere Business Monitor server to ensure that all the changes are applied.

8.6.3 Exercising the WebSphere MQ Workflow process

Now we use the WebSphere MQ Workflow Client to run instances of a WebSphere MQ Workflow process so that we can see the monitoring result later from the WebSphere Business Monitor dashboards. To exercise the WebSphere MQ Workflow process:

1. Make sure the WebSphere MQ Workflow server is started.
2. Check the Windows services application and make sure IBM MQ Workflow 3.6 - FMC service is started.
3. Open WebSphere MQ Workflow client for FMC configuration. Click **Start** → **All Programs** → **IBM** → **IBM WebSphere MQ Workflow** → **WebSphere MQ Workflow Client - FMC**.
4. Log in with the user ID and password. The initial user ID is `ADMIN` and password is `password`.

Archived

5. Configure the WebSphere MQ Workflow client to show the process templates. In the Tree View, right-click **Process Template Lists** and select **Create New Process Template List...**.
6. Similarly create the Process Instance List and Worklist.
7. In the Process Templates pane (Figure 8-142), right-click the **OrderHandling** process and click **Create and Start Instance**.

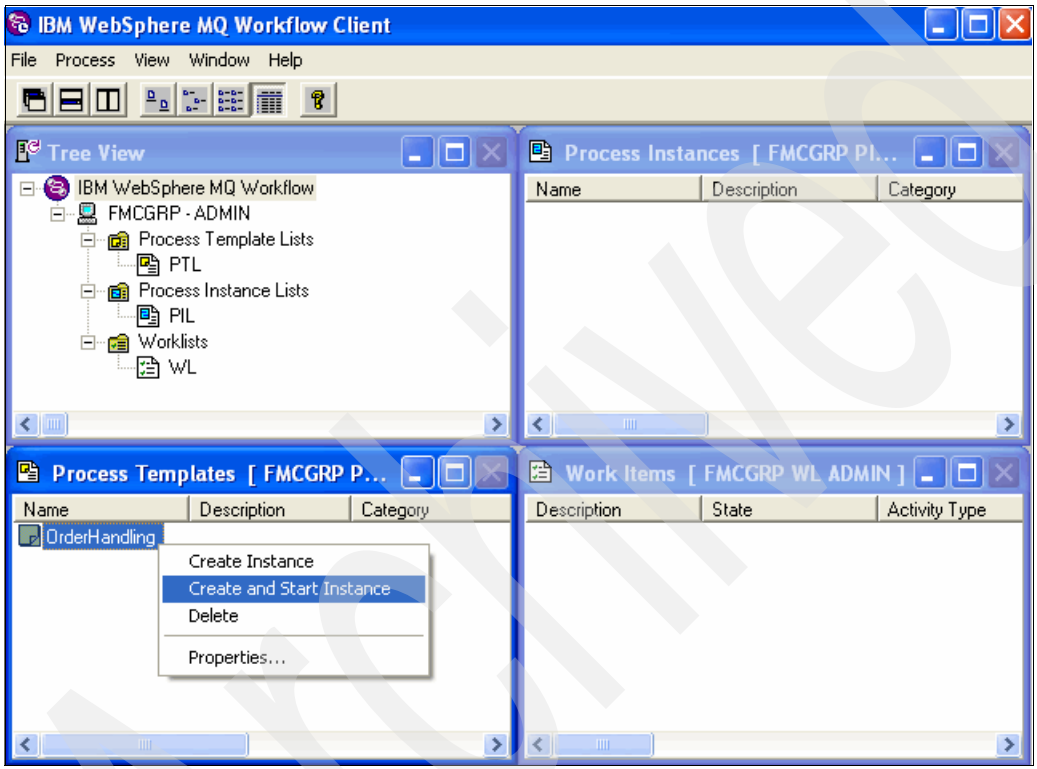


Figure 8-142 WebSphere MQ Workflow client

Archived

8. In the Input data structure [Order] window (Figure 8-143), type input values for the Order data members and click **OK**. The default values are provided, so when we start a instance, the necessary values are set. We can directly use them and change values as needed.

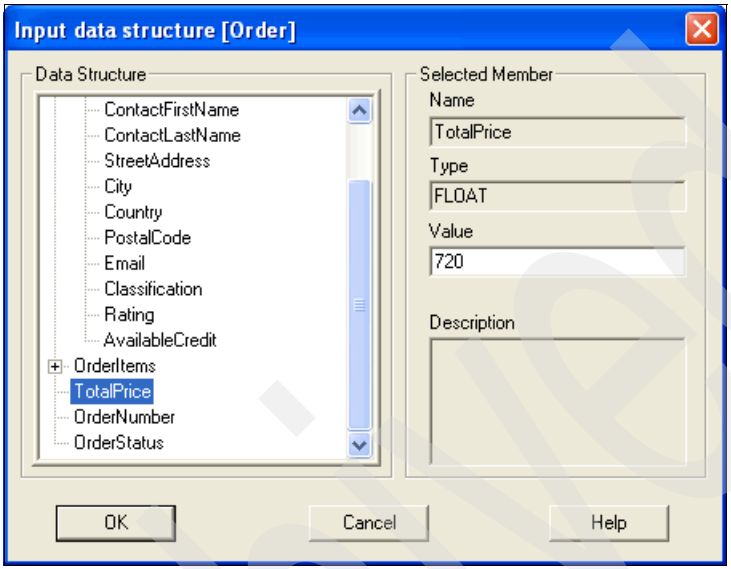


Figure 8-143 Values for data members

Archive

9. A process instance and work item are created. Refresh the Process Instances and Work Items window (Figure 8-144) to view them.

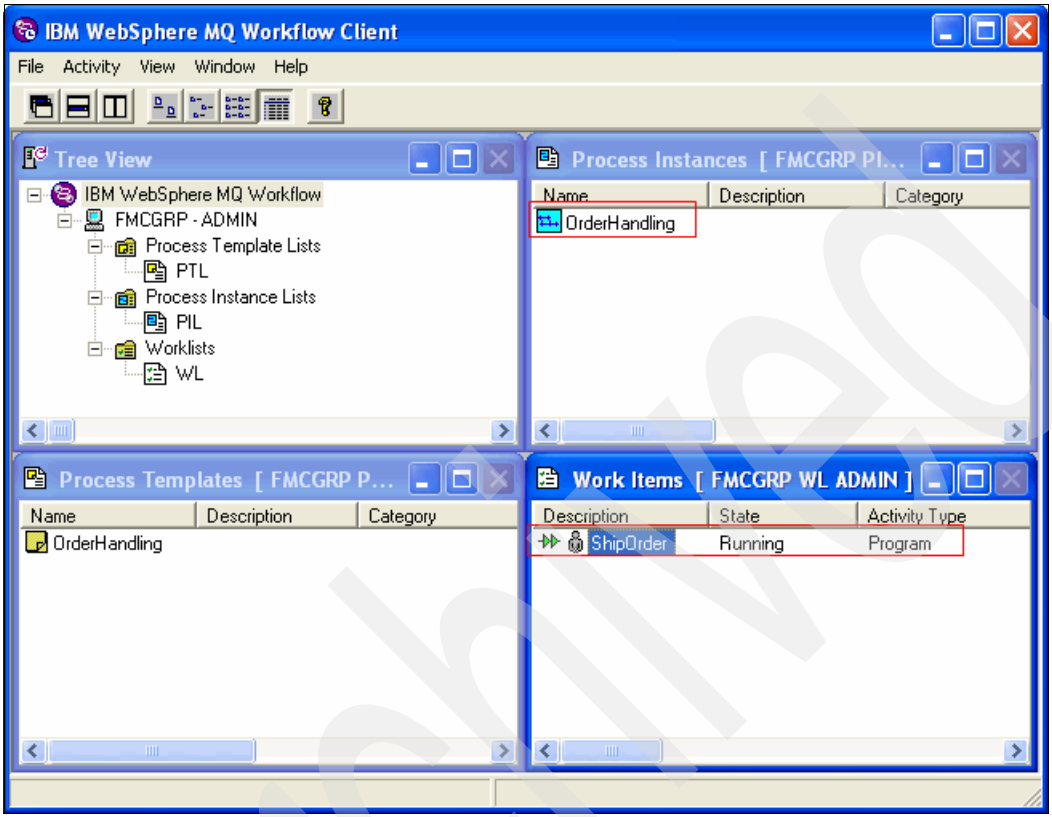


Figure 8-144 Process instance and work item are created

10. When the process instance goes into a human task, the IBM WebSphere MQ Workflow - Show Container window opens (Figure 8-145 on page 585). Steps a, d, f, and g are required to accomplish the task.

    a. Click the **Get Data** button.

    b. Ensure that Process Model is `OrderHandling`.

    c. Ensure that Activity is `ShipOrder` or `ReviewOrder`.

    d. Click the **--->** button to set the output data.

    e. Change the output value from the right pane.

    f. Click the **Set Output Data** button to set the changes of data to the output container.

    g. Click **Exit** to finish the human task.

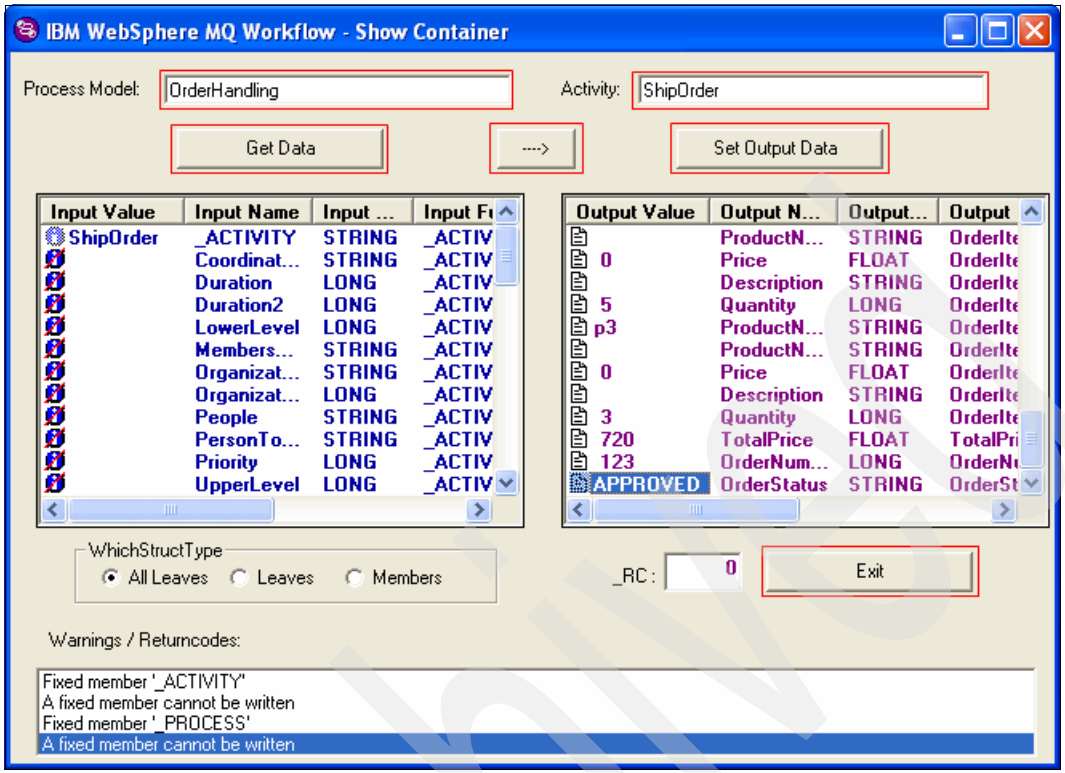


Figure 8-145 Setting the human task output

Run additional instances by using the previous steps and change the input data and output data as needed to make instances go through different paths of the process. Table 8-5 contains example input and output values for reference. We also must set the different Customer.City and Customer.Country values so that we can see the statistic data breakdown by different locations in the Dimensions view.

Table 8-5 Example input values

Instance	Input values when starting an instance	Output values for the ReviewOrder activity
1	Customer.Classification=GOLD Customer.AvailableCredit=10000 Total Price=1500	
2	Customer.Classification=GOLD Customer.AvailableCredit=1000 Total Price=1500	OrderStatus=APPROVED

Instance	Input values when starting an instance	Output values for the ReviewOrder activity
3	Customer.Classification=SILVER Customer.AvailableCredit=2000 Total Price=1500	OrderStatus=DECLINED
4	Customer.Classification=NORMAL Customer.AvailableCredit=5000 Total Price=1000	OrderStatus=APPROVED
5	Customer.Classification=NORMAL Customer.AvailableCredit=5000 Total Price=500	

8.6.4 Configuring the dashboard views

Now we configure the WebSphere Business Monitor Web dashboard to show the monitoring result:

1. Open the dashboard (Figure 8-146) and use `admin` to log in, which is the user ID that we set for dashboard alerts.

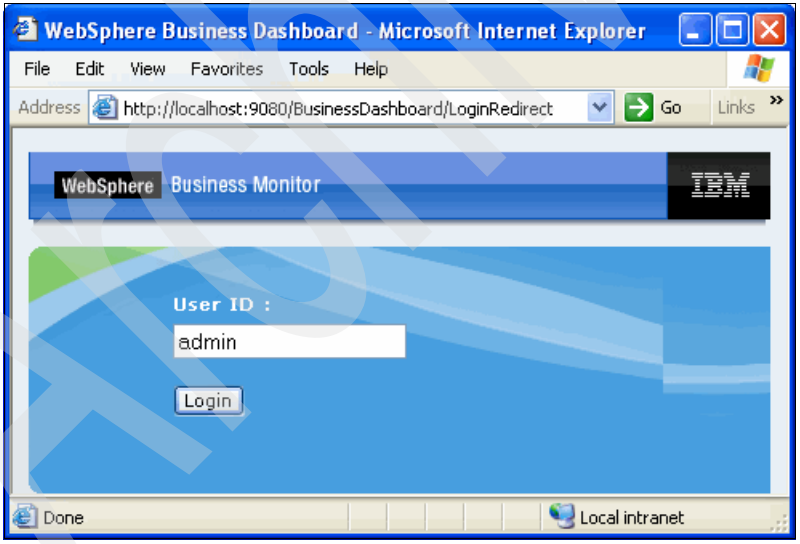


Figure 8-146 Login dashboard

2. On the Getting Started page, click the **Dashboards** tab to switch to the Dashboards page.

3. On the Dashboards page (Figure 8-147), under the **Manage** tab, click **New** to create a new dashboard.

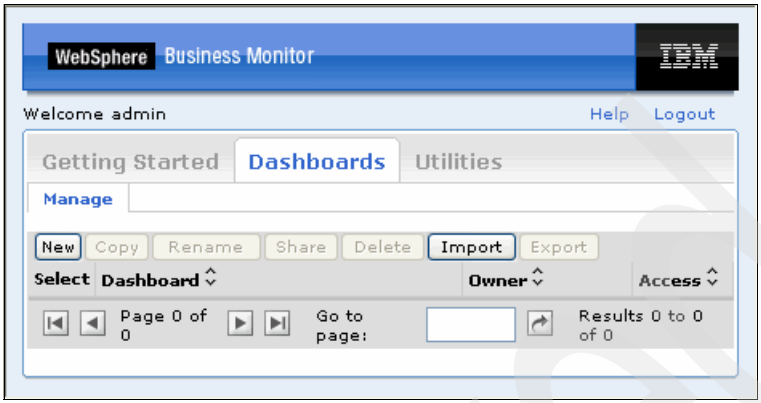


Figure 8-147 Dashboards page

4. In the New Dashboard panel (Figure 8-148), for Name, type `OrderHandling`, and click **OK**.

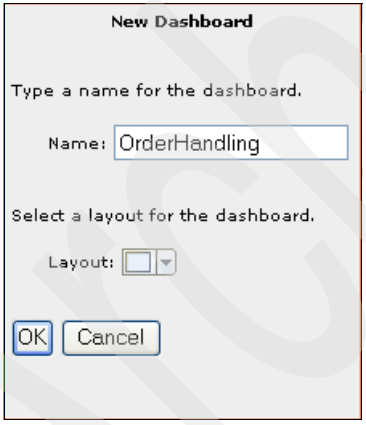


Figure 8-148 Setting the new dashboard name

Archived

5. On the OrderHandling page (Figure 8-149), click **Add to Dashboard** to add views.

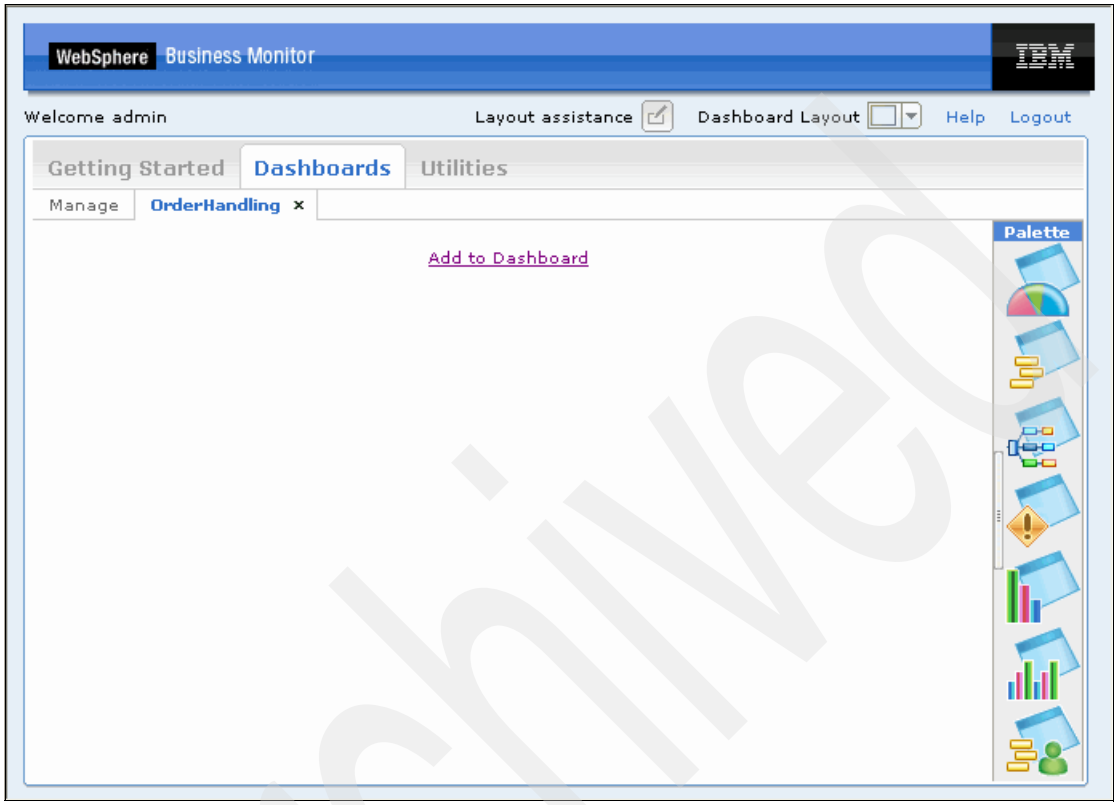


Figure 8-149 OrderHandling page

6. In the Add to Dashboard panel (Figure 8-150), select **Diagrams** and click **OK**. The Diagrams view is added to OrderHandling page.

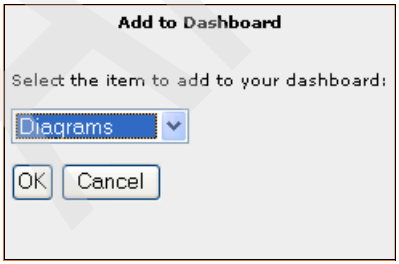


Figure 8-150 Selecting a view

Archived

7. On the Diagrams page (Figure 8-151), click **Personalize** to configure the view.

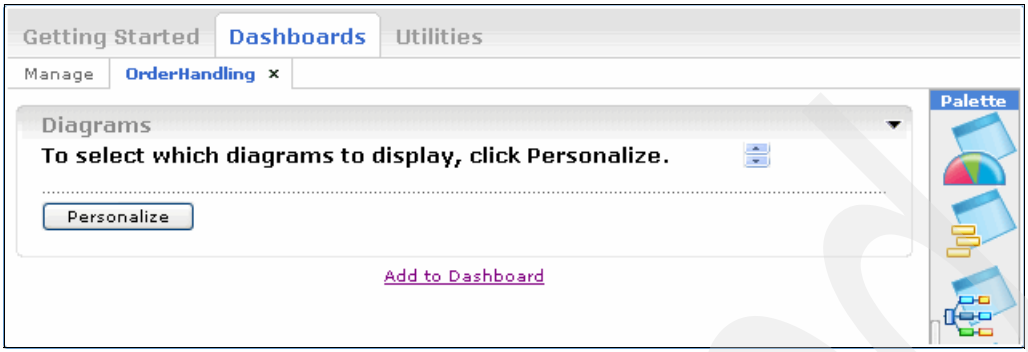


Figure 8-151 Initial Diagrams view

8. Configure the Diagrams view:
   a. In the Select a Diagram page (Figure 8-152), for Model, select **OrderHandlingMM**. For Context, select **Key Performance Indicator context**, and then select **OrderHandlingKC**.

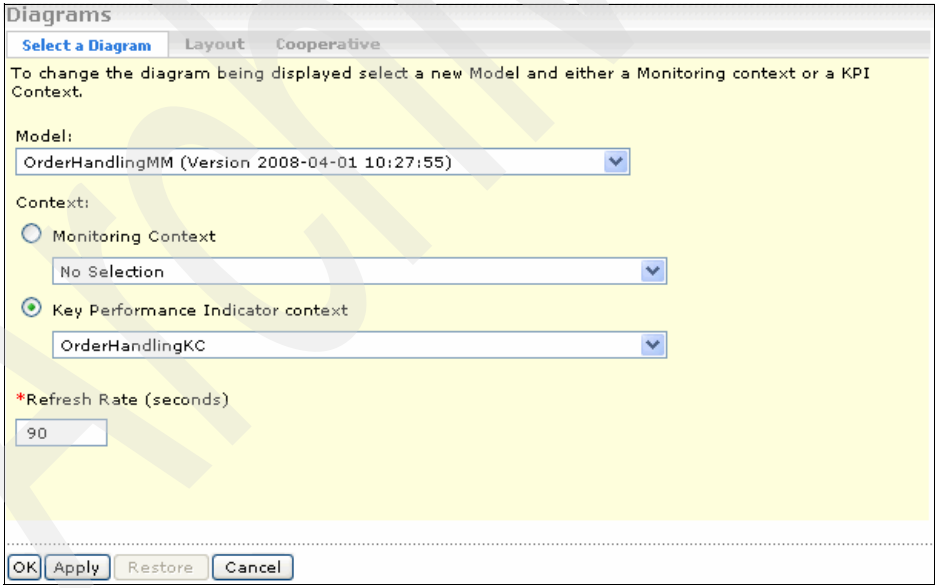


Figure 8-152 Selecting the KPI context diagram

b. Click the **Cooperative** tab. On the Cooperative page (Figure 8-153), select **Cooperative mode**.

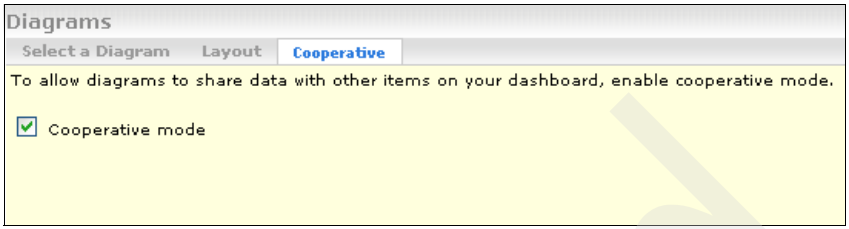


Figure 8-153 Enabling cooperative mode in the Diagrams view

c. Click **OK** to apply the configuration. Figure 8-154 shows the diagram.

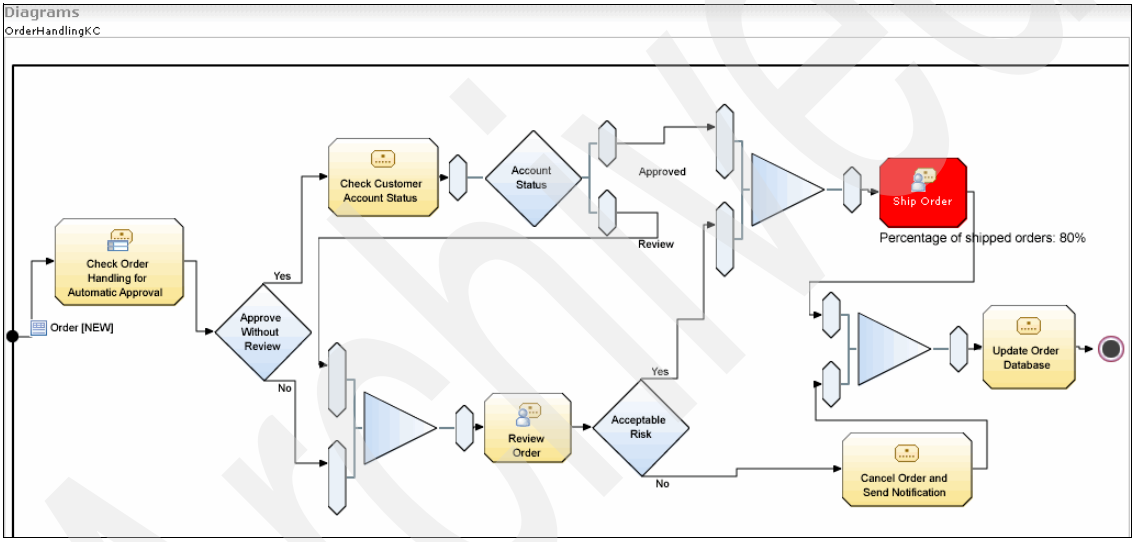


Figure 8-154 Diagrams view

Archived

9. Similarly add an Instances view to the dashboard by using the following configuration:

   a. On the Show/Hide page (Figure 8-155), under Available, select **COMPLETED**, **Order Fulfillment Duration**, **City**, **Country**, **Order Number**, **Total Price**, and **Order Status**. Then click **>** to move the columns to the Selected box.

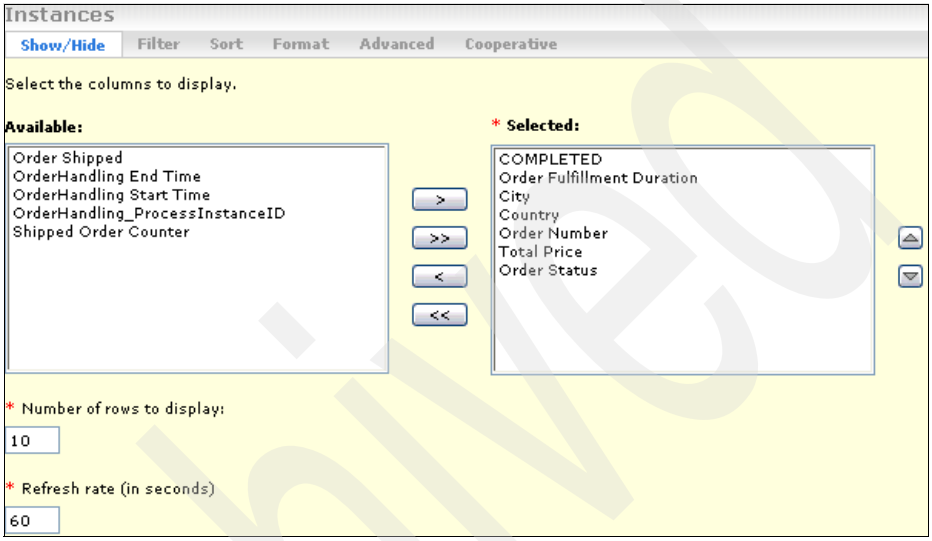


Figure 8-155 Selecting the basic instance columns to display

   b. Click the **Cooperative** tab (Figure 8-156). Select **Cooperative mode**.

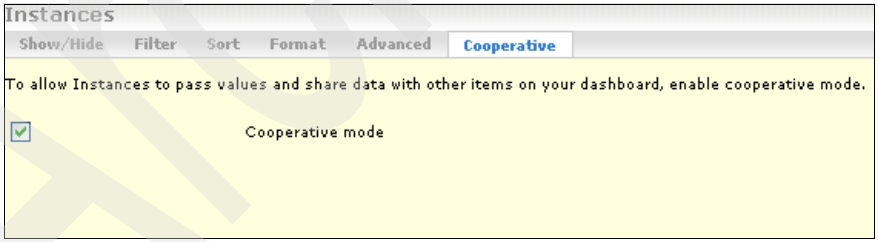


Figure 8-156 Enabling the cooperative mode in the Instances view

   c. Click **OK**.

Archived

10. In the Instances view (Figure 8-157), click the icon under the Diagram column for one instance.

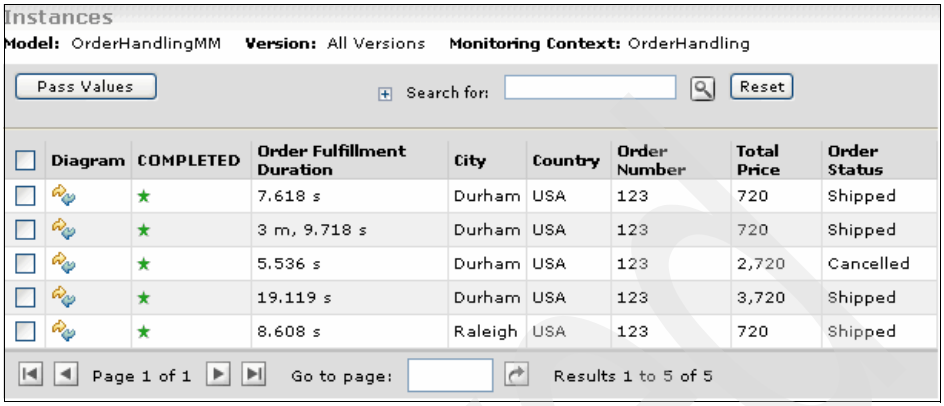


Figure 8-157 Instances view

11. Check the Diagrams view (Figure 8-158), which shows the status of the specified instance.

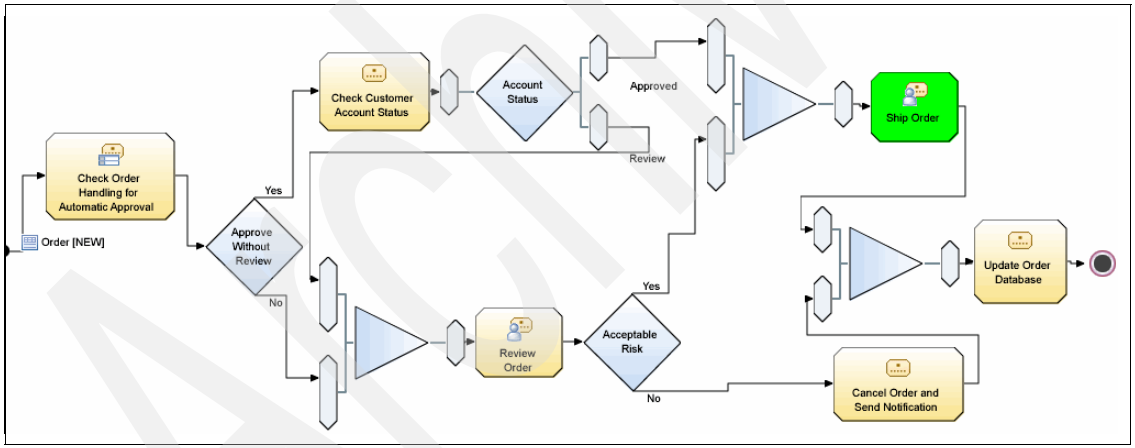


Figure 8-158 The diagram linked with one instance

Archive

12. Add the KPIs view to the dashboard and configure it by clicking **Personalize**. On the Select KPIs page (Figure 8-159):

    a. Select **All versions**.

    b. Select the **Average Order Fulfillment** and **Percentage of Shipped Orders** KPIs.

    c. Click **OK**.

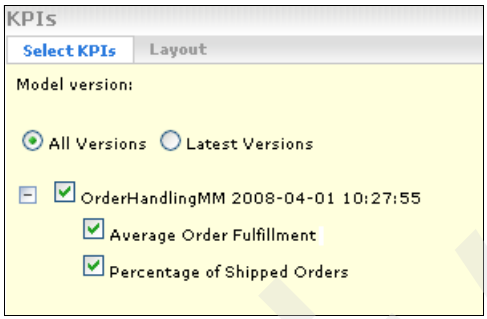


Figure 8-159 Selecting the KPIs to display

Figure 8-160 shows the KPIs view.

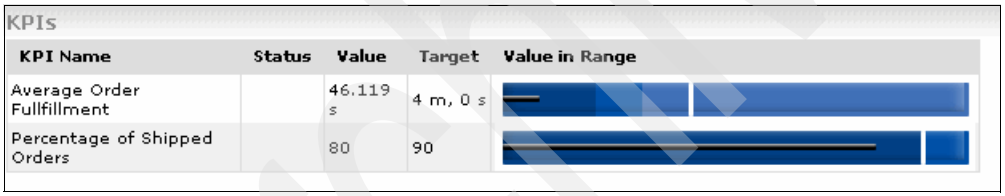


Figure 8-160 KPIs view

Archived

13. Add the Dimensions view to the dashboard and configure it by clicking **Personalize**. In the Dimensions view (Figure 8-161):

    a. For Monitoring Model, select **OrderHandlingMM**.
    b. For Monitoring context, select **OrderHandling**.
    c. For Row dimensions, select **Location**.
    d. For Column dimensions, select **Measures**.
    e. Click **OK**.

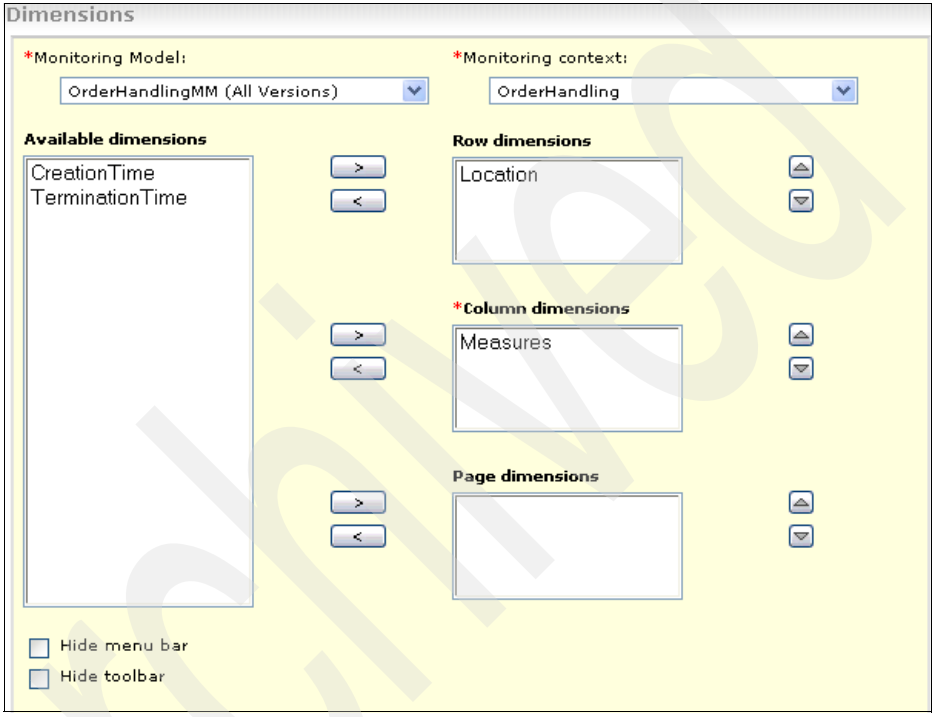


Figure 8-161 Setting the dimensions for the Dimensions view

Archive

14. In the initial Dimension view (Figure 8-162), right-click **InstancesCount** and click **Member Filter...**.

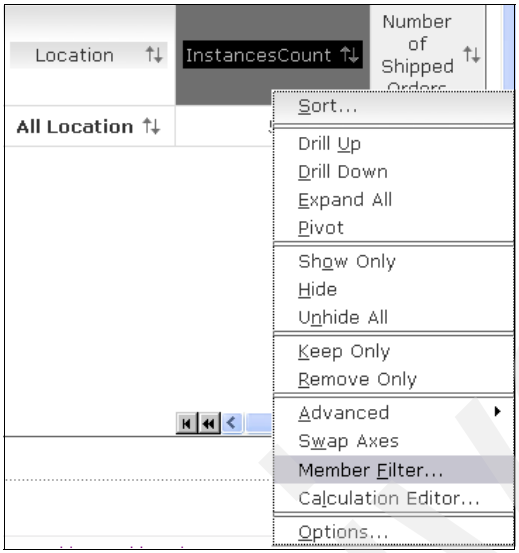


Figure 8-162 Selecting the Member Filter menu

15. In the Member Filter -- Web Page Dialog (Figure 8-163), under Selected Members, select **InstancesCount** and click **Remove**. Then click **OK**.

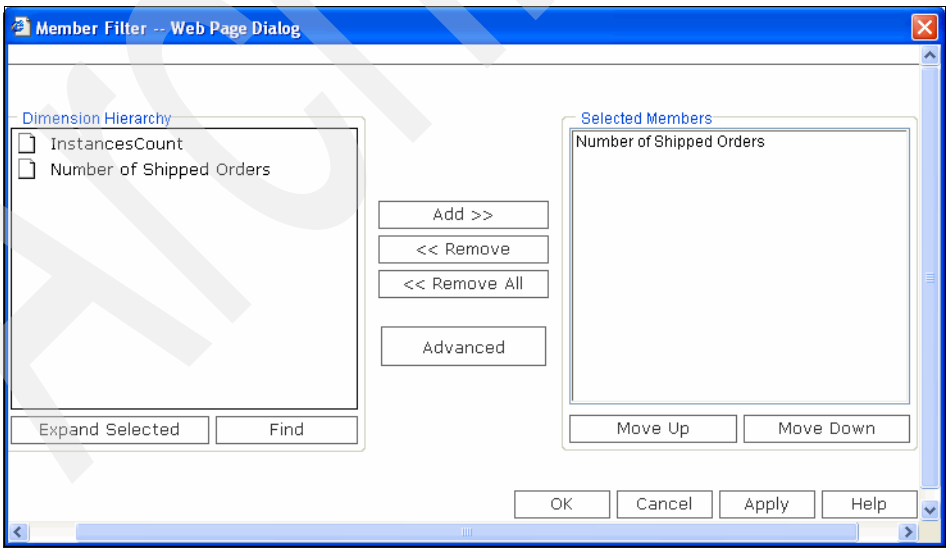


Figure 8-163 Only Number of Shipped Orders selected

Archived

Figure 8-164 shows the Dimensions view.

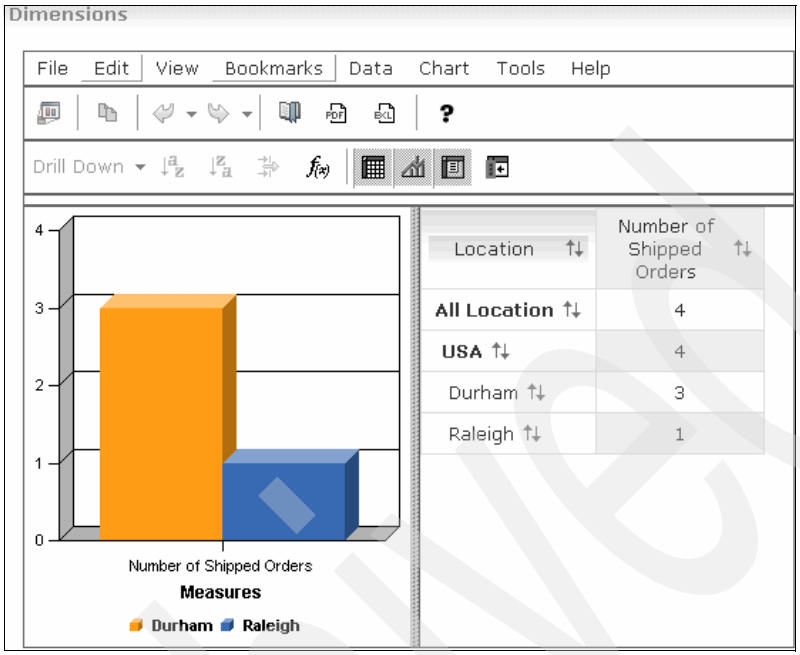


Figure 8-164 Dimensions view

16. Add the Alerts view (Figure 8-165) to the dashboard.

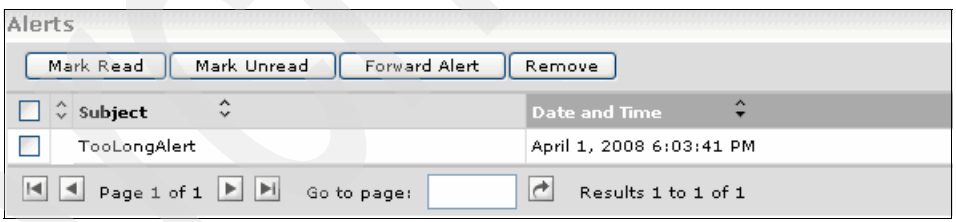


Figure 8-165 Alerts view

Archive

17. On the Alerts page (Figure 8-166), under the Subject column, click the alert and check the alert details.

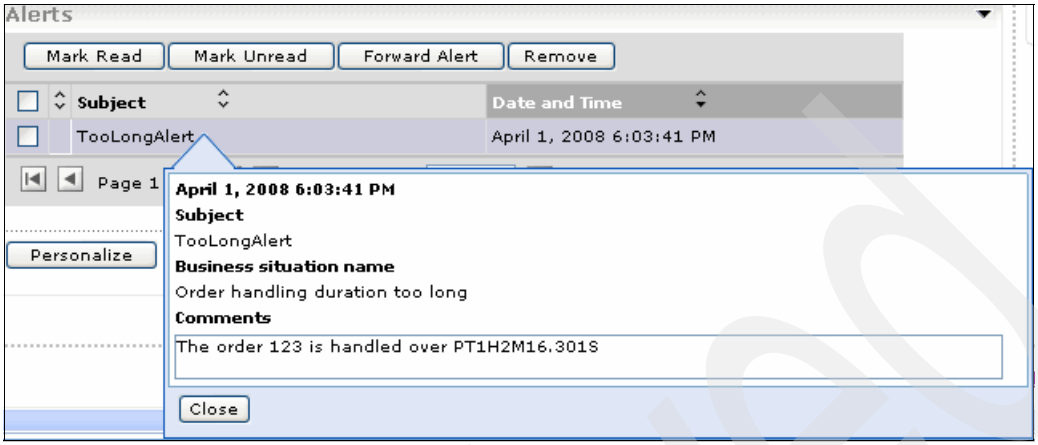


Figure 8-166 Alert details

18. Explore the dashboards by generating more WebSphere MQ Workflow process instances.

8.7 Summary

In this chapter, we described the architecture of WebSphere MQ Workflow support in WebSphere Business Monitor V6.1 and introduced an end-to-end sample.

We set up the runtime and development environments, implemented the WebSphere MQ Workflow process and monitor model, and deployed the necessary configurations. Finally, we tested the developed monitor model with the WebSphere Business Monitor dashboard views.

Archived

Archived

9

Custom event emitter

The Common Event Infrastructure (CEI) serves as an integration point for the consolidation and persistence of raw events. Such raw events range from multiple heterogeneous sources and the distribution of Common Base Event formatted events to event consumers such as WebSphere Business Monitor.

IBM applications, such as WebSphere Process Server, WebSphere Enterprise Service Bus (ESB), and WebSphere MQ Workflow, have the capability to generate and emit Common Base Event formatted events. However, not all applications have the built-in capability for creating an event or an emitter for sending events. An example of this is a custom developed application.

In this chapter, we discuss how we can enable these applications for business activity monitoring (BAM).

© Copyright IBM Corp. 2008. All rights reserved.

9.1 Monitoring arbitrary events

Figure 9-1 shows the general flow of events from event source to event consumer by using the CEI.

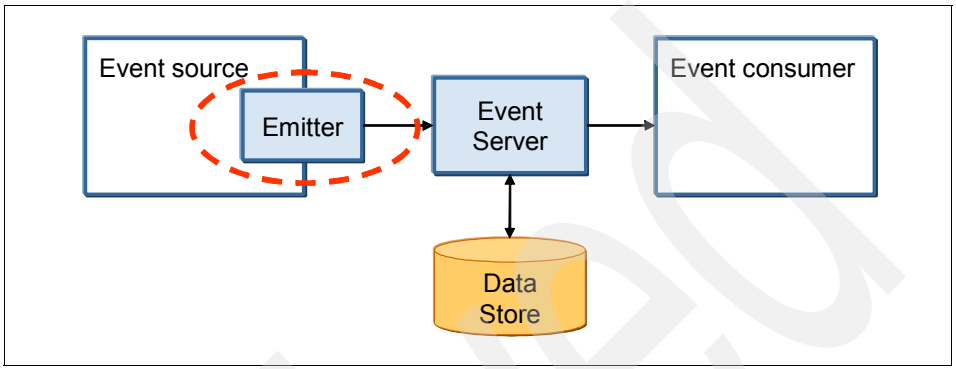


Figure 9-1 Common Event Infrastructure

Some applications have a built-in emitter component. In this section, we focus on the emitter component. We discuss the options that are available for applications that do not have a built-in emitter component and do not generate Common Base Event formatted events.

9.1.1 Approaches for emitting Common Base Events

There are several approaches for emitting events to be consumed by WebSphere Business Monitor where an emitter component is not available:

- Using Adapter and ESB technologies to generate events on behalf of the monitored application (Refer to Chapter 6, “WebSphere ESB and IBM Adapters” on page 249.)
- Modifying the existing application directly to emit Common Base Events that contain the required information

  a. The events are used on a messaging infrastructure, such as WebSphere MQ, to route text messages in the XML format to the server queue of the CEI server.

  b. The CEI server forwards these events to the appropriate monitor models.

Archived

- Writing a stand-alone application that acts as the event emitter
  a. The applications detects changes or the occurrence of an activity on the event source (for example, the update of database tables) and receives information about these events.
  b. The application uses the Common Base Event and the CEI APIs to create and emit events in the Common Base Event format for consumption by WebSphere Business Monitor.

We discuss the first approach of using Adapters and ESB in Chapter 6, "WebSphere ESB and IBM Adapters" on page 249. In 9.8.1, "Custom event emitters and WebSphere Adapters" on page 661, we compare the approach of using Adapters and ESB against developing custom event emitters.

In Chapter 10, "WebSphere MQ" on page 665, we discuss the approach of emitting Common Base Events from an existing application and leveraging on WebSphere MQ to route these messages to the server queue of the CEI Server.

In this chapter, we discuss the approach of writing a stand-alone event emitter.

9.2 Event emitter concepts

In this section, we discuss briefly general concepts related to Common Base Event and CEI as well as the development of event emitters in general. In the following sections, we illustrate how these concepts are used in the development of a sample custom event emitter.

9.2.1 Using the emitter framework

To facilitate the development of custom event emitters, a simple emitter framework is made available. This is a common framework used by the sample event emitters available at the following Web address:

http://www-306.ibm.com/software/integration/wbimonitor/library/tutorials.html

The sample file event emitter discussed here uses this event emitter framework as well.

Note: You can develop your own emitter framework subsequently, but the framework is provided here for consistency and serves as a helpful tool to get started.

Figure 9-2 shows a class diagram of this framework.

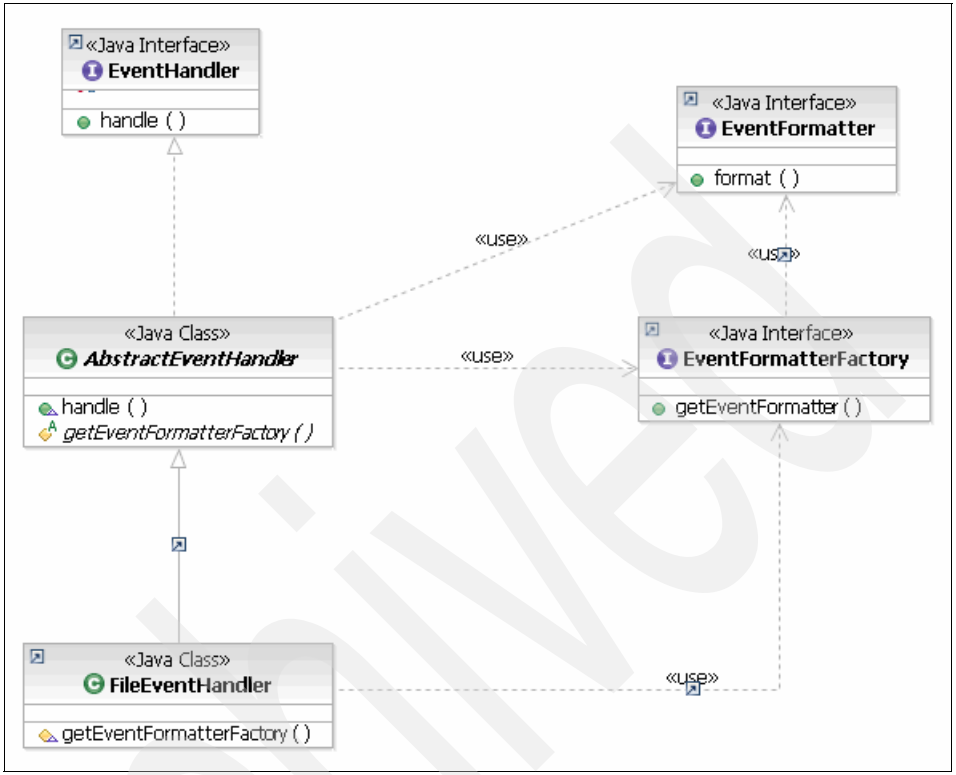


Figure 9-2 Event emitter framework

Event emitter framework flow

The following flow is regulated by this common framework when an event in the enterprise backend system is detected:

1. From the EmitterFormatterFactory, retrieve an EventFormatter specific for the type of data being processed.
2. Invoke the EventFormatter to convert the input data to a Common Base Event object.
3. Retrieve an emitter from an EmitterFactory to send the event to the CEI server.
4. Emit the Common Base Event to the CEI server.

In the following sections, we discuss the steps in the event emitter framework flow.

Archived

9.2.2 Retrieving EventFormatter from the EmitterFormatterFactory

Before we can emit a Common Base Event to the CEI server, we must convert data from the event source in its existing format to the Common Base Event format. To do this, we first retrieve, from the EmitterFormatterFactory, an event formatter that is specific to the type of data that is processed. We see this in Example 9-1.

Example 9-1 Retrieving the EventFormatter from the EmitterFormatterFactory

```
//Get an appropriate EventFormatter and format the event.
EventFormatter formatter =
getEventFormatterFactory().getEventFormatter(o);
```

9.2.3 Invoking the EventFormatter or creating Common Base Events

After obtaining the EventFormatter, we invoke it as shown in Example 9-2 to create a Common Base Event from the formatted input data.

Example 9-2 Invoking the EventFormatter

```
//Invoke the EventFormatter
CommonBaseEvent event = formatter.format(o);
```

Generating Common Base Events: In the sample in this chapter, we use the EventFormatter to convert events from an existing data type to the Common Base Event. However, in some situations, you may prefer to generate the Common Base Event directly. For example, you might modify your existing application to emit Common Base Events instead of creating a separate stand-alone application for this.

In this case, instead of using the EventFormatter that is available with the Event Emitter Framework, you can choose to create a Common Base Event yourself. Both approaches use the Common Base Event APIs.

Creating a new or obtaining an existing EventFactory

Example 9-3 shows how to create a new EventFactory with a ContentHandler.

Example 9-3 Creating a new EventFactory with a content handler

```
//Create a new EventFactory using the EventFactoryFactory
EventFactory eventFactory = (EventFactory)
EventFactoryFactory.createEventFactory(contentHandler);
```

Example 9-4 shows how the code fragment looks if a ContentHandler is not used.

Example 9-4 Creating a new EventFactory without a content handler

```
//Create a new EventFactory using the EventFactoryFactory
EventFactory eventFactory = (EventFactory)
EventFactoryFactory.createEventFactory()
```

Content handler: A content handler is an object that automatically sets the property values of each event based on any arbitrary policies that you want to use. It provides automatic content completion for newly created event.

We can also obtain an existing EventFactory by using the Java Naming and Directory Interface (JNDI). It inherits the ContentHandler, if one exists.

Example 9-5 Obtaining an existing EventFactory by using JNDI

```
//Obtain an existing EventFactory
Context context=new InitialContext();
EventFactory eventFactory = (EventFactory)
context.lookup("com/ibm/events/EventFactory");
```

Policy compliance of event sources: To ensure that all event sources comply with the same policies, you can create an event factory associated with a content handler (by using EventFactoryFactory). Then you can bind the created event factory into a JNDI namespace. Instead of creating their own event factories, event sources can then perform JNDI lookups to access the event factory that exists, without any knowledge of the content handler. If your business rules change later, you can modify the content handler in one place.

Creating Common Base Events

Example 9-6 shows the creation of a Common Base Event by using the eventFactory.

Example 9-6 Creating a Common Base Event

```
//Creating Common Base Event
CommonBaseEvent event =
eventFactory.createCommonBaseEvent("ActivityEvent")
```

9.2.4 Retrieving an emitter from an EmitterFactory

An *emitter* is a local object that provides methods for sending events. An event source does not interact directly with the event service to send events. Instead, it interacts with an emitter.

In general, the emitter handles the details of an event transmission. The developer of an event source does not need to be concerned about the event service location, the filter settings, or the underlying transmission mechanism. Details such as this are governed by the EmitterFactory and an object configured by an administrator. They are bound in a JNDI namespace.

An *emitter factory* is an instance of com.ibm.events.emitter.EmitterFactory and is used to create emitter objects. It also defines the behavior of the emitters it creates. In Example 9-7, we retrieve an emitter from the an EmitterFactory after we create a Common Base Event.

Example 9-7 Obtaining the EmitterFactory

```
import javax.naming.*;
import com.ibm.events.*;
Context context=new InitialContext();
EmitterFactory emitterFactory =
(EmitterFactory)context.lookup("com/ibm/events/configuration/emitter/De
fault")
```

9.2.5 Emitting the Common Base Event to the CEI server

An emitter performs several kinds of processing before sending the event to the event service. To send an event, you use the sendEvent() methods of the emitter interface. When you submit an event to an emitter, the following steps occur:

1. The emitter calls the complete() method of the event, triggering optional content completion.
2. The emitter assigns a sequence number and global instance identifier to any event that does not already have them.
3. The emitter validates the event to ensure that it conforms to the Common Base Event specification.
4. If filtering is active, the emitter checks the event against the current filter criteria to determine whether the event should be sent or discarded.
5. If the event is valid and passes the filter criteria, the emitter sends the event to the event service for persistence and distribution to the event consumers.

Archived

If the event is not valid, or if the emitter encounters a problem when trying to send the event to the event service, an exception is thrown.

Example 9-8, shows the emission of Common Base Event events to the CEI server.

Example 9-8 Obtaining an emitter from the EmitterFactory and sending an event

```
//Get an emitter and send the event
emitter=emitterFactory.getEmitter();
emitter.sendEvent(event);
```

9.3 Scenario description

In this chapter, we show the development of a stand-alone application to interact with the application or applications to be monitored and emit events that contain pertinent information for business activity monitoring. We use the file event emitter sample in our discussion.

The goals of this section are to introduce the following elements through the fictitious scenario ClipsAndTacks:

- The libraries and APIs provided by the CEI to generate and emit business events in the form of a Common Base Event
- The required steps for setting up an event emitter application for BAM

In this scenario, ITSO Logistics is a major international shipping company. ITSO Logistics has an existing contract with ClipsAndTacks Office Supplies for it to handle ClipsAndTacks' shipping needs, shipping to ClipsAndTacks' customers across more than 100 countries worldwide.

As part of their existing agreement, ITSO Logistics provides a shipping notification to ClipsAndTacks for each order that has been shipped (Figure 9-3).

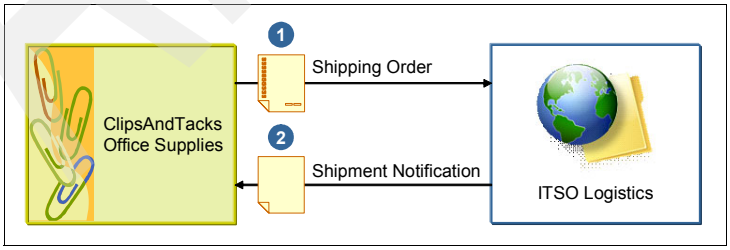


Figure 9-3 ITSO Logistics

This notification file details the product number, quantity, and the cost and days it took to ship the product (Example 9-9).

Example 9-9 Sample shipment notification file

```
TYPE=ORDER
EVENTACTION=CREATE

OrderID=012345
CustID=011011
Status=SHIPPED
Country=KOREA
ProductID=JK1234
Quantity=3
TotalPrice=600
ShipmentCost=20
ShipmentTime=3
```

The agreed mode of communication between ITSO Logistics and ClipsAndTacks is to use File Transfer Protocol (FTP). ClipsAndTacks sends a request for the order to be shipped, and ITSO Logistics sends a shipment notification file on a designated folder in an FTP server after the order is shipped or rejected.

ClipsAndTacks wants to monitor the average turnaround time of these orders and the average cost it took to ship to their destination.

External event source: This is an example of a scenario where the event sources are not managed by a process choreographer such as WebSphere Process Server. In situations where event sources are managed, the process engine may be instrumented to emit events as they occur. However in this case, the event source is outside ClipsAndTacks' organization and such an instrumentation may not be possible.

9.4 Design overview of the emitter application

The file event emitter is a J2EE enterprise application that also implements the WebSphere TaskHandler interface, allowing it to handle invocations from a WebSphere Application Server Scheduler. We describe the overall event retrieval and event emission flow as shown in Figure 9-4 on page 608.

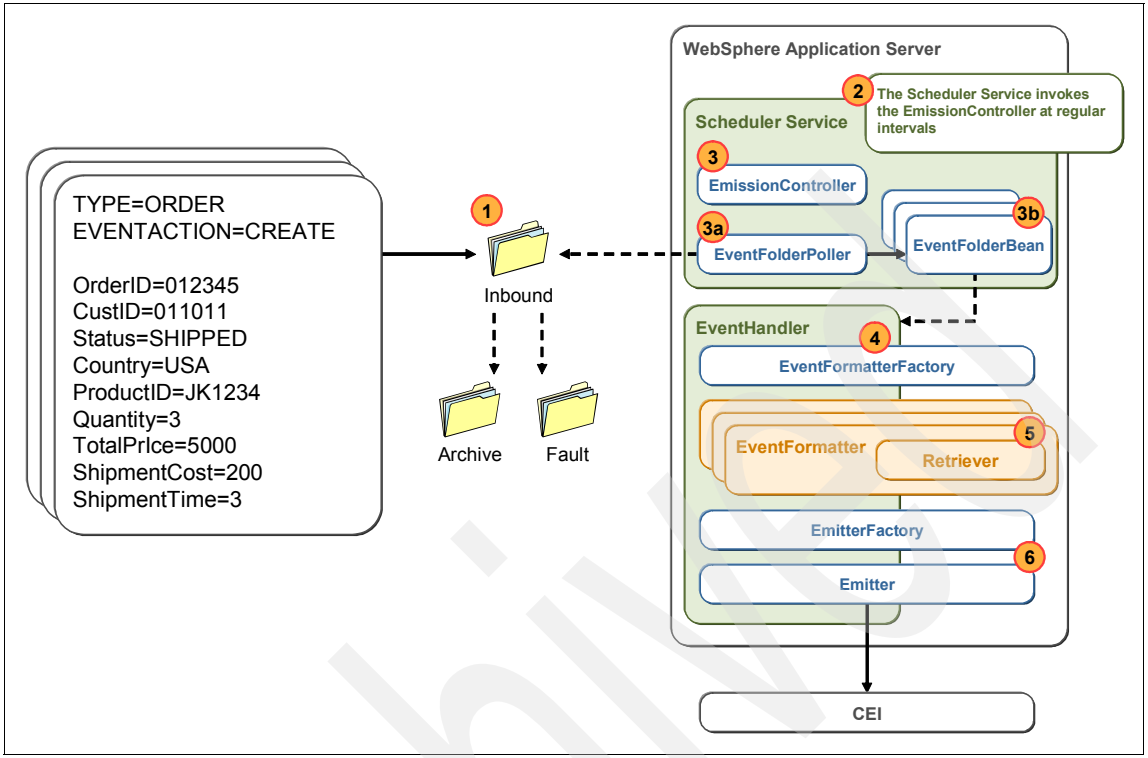


Figure 9-4 Event emitter flow

The flow follows this sequence:

1. A file encapsulating information about a create, update, and delete event is placed in the inbound directory. This file contains information about the data type that the event pertains to, the event type (that is, create, delete, or update), and the attributes and their associated values.
2. The Scheduler service in WebSphere Application Server invokes the EmissionController at a specified interval.
3. The EmissionController calls EventFolderPoller.
   a. If any files are found in the inbound directory, EventFolderPoller populates and returns a collection of EventFileBean objects.
   b. If the collection of EventFileBean objects is not empty, EmissionController invokes the handle() method of EventHandler for each EventFileBean object to handle the emission steps.
4. EventHandler calls the EventFormatterFactory, which returns an EventFormatter object according to the application data type specified in the event file.

5. The EventFormatter object transforms the EventFileBean object into a CommonBaseEvent object.
6. The CEIEmitter sends the Common Base Event object to the CEI event server.

9.5 Importing and working with the source code project

The File Event Emitter sample project interchange is in the additional materials for this book in the /eventemitter/FileEmitter_beforePI.zip file. For more information about how to obtain these materials, see Appendix B, "Additional material" on page 801. Table 9-1 lists the contents of the modules in the project.

Table 9-1 File Event Emitter project descriptions

Project name	Description
Emitter FW	The common emitter framework that is used by all the sample event emitters
CEIEmitter	The CEI emitter code, which is also the common code shared by all the sample event emitters
FileEmitterEAR	The enterprise application package that is deployed to the server
FileEmitterUtil	File emitter code shared by FileEmitterEJB and FileEmitterImpl
FileEmitterEJB	File emitter specific EJB code
FileEmitterImpl	File emitter specific code This project contains the implementation code for retrieving files from the file system and formatting these events into Common Base Events.

Prerequisites

Before you proceed, ensure that the WebSphere Business Monitor Toolkit is installed on the Eclipse platform that you are working on. Also ensure that you have a monitor profile running on either WebSphere Application Server, WebSphere Enterprise Service Bus, or WebSphere Process Server.

This example was created using the following setup:

- Rational Application Developer v7.0.0.6
- WebSphere Application Server v6.1.0.15
- DB2 Enterprise Server Edition Version 9.1
- WebSphere Business Monitor Toolkit v6.1.1.0
- IBM Installation Manager V1.1.1

Archived

Note: In our scenario, we deploy the emitter application to the same application server as the WebSphere Business Monitor server. In your implementation, you can also consider deploying the emitter applications to its own application server.

Creating the inbound, fault, and archive folders

We use the default folder locations C:\FileEmitter\inbound, C:\FileEmitter\fault, and C:\FileEmitter\archive. Create these folders and note the directories in which you create them. These folders are used for event management.

9.5.1 Importing the project

To import the project:

1. Launch Rational Application Developer. Close the Welcome page.
2. From the menu bar, select **File** → **Import**.
3. In the Import – Select window (Figure 9-5), expand **General** and select **Existing Projects into Workspace**. Then click **Next**.

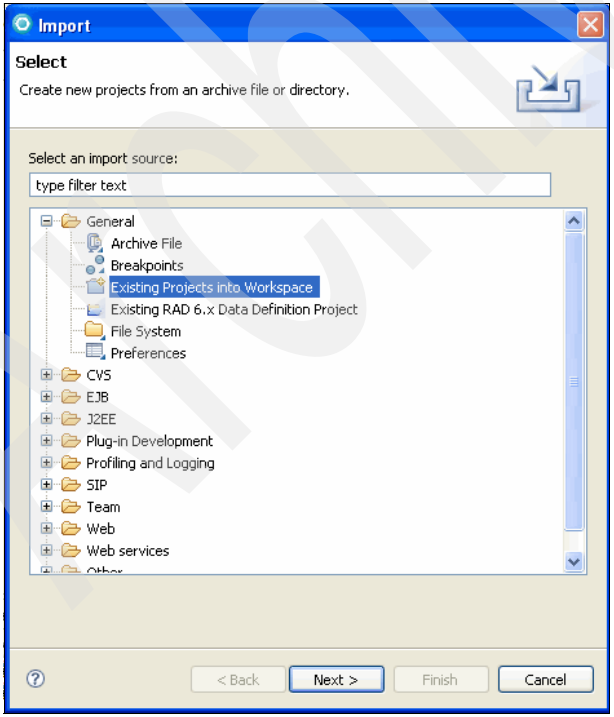


Figure 9-5 Importing existing projects into the workspace

4. In the Import – Import Projects window (Figure 9-6):
   a. Click the **Select archive file** option. Then click **Browse**.
   b. Look for the project interchange file. Select **FileEmitter_beforePI.zip** and click **Open**.
   c. Under Projects, ensure that all the Projects are selected.
   d. Click **Finish**.

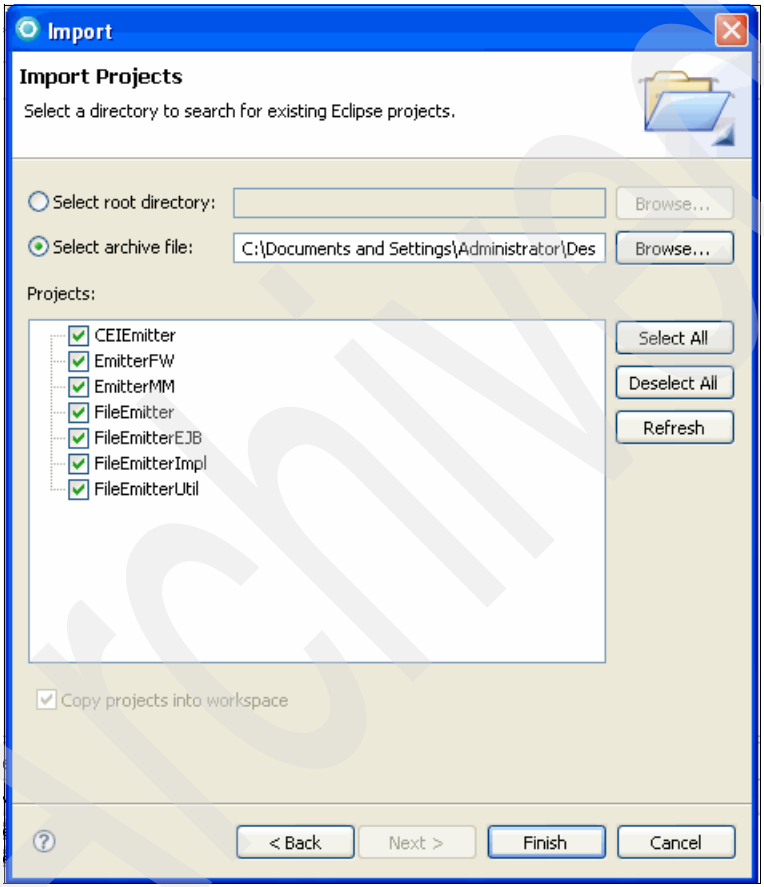


Figure 9-6 Importing the archive file

Archive

You can expect to see build error messages in the Problems view after importing the project interchange file. To resolve this error message, modify the classpath reference to the com.ibm.events.clients_6.1.0 JAR plug-in in the CEI Emitter project:

1. Right-click the **CEIEmitter** project and select **Properties**.
2. In the Properties for CEIEmitter window (Figure 9-7), in the left pane, select **Java Build Path**. In the right pane, click the **Libraries** tab. Select **WAS_61_PLUGINDIR/com.ibm.events.clients_6.1.0.jar** and click **Edit**.

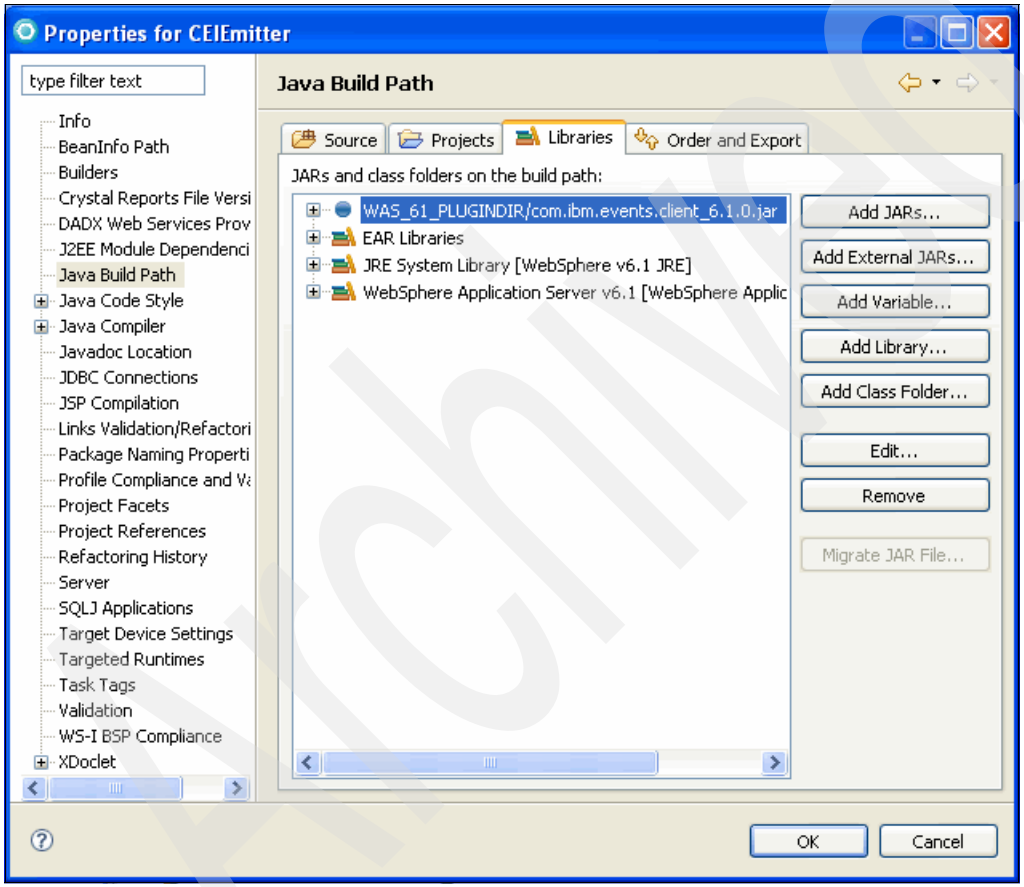


Figure 9-7 Java Build Path pane of the Properties for CEIEmitter window

3. In the Edit Variable Entry window (Figure 9-8), click **Variable**.

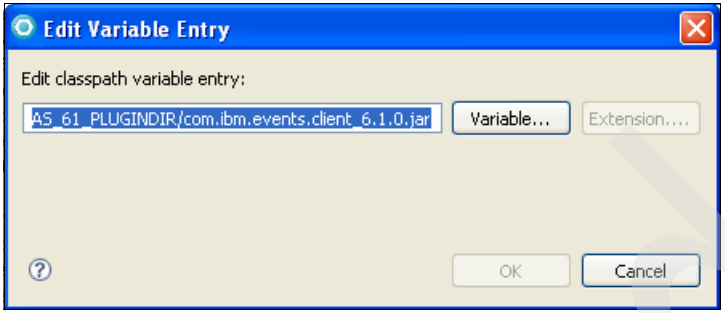


Figure 9-8 Edit Variable Entry window

4. In the Variable Selection window (Figure 9-9), click **New**.

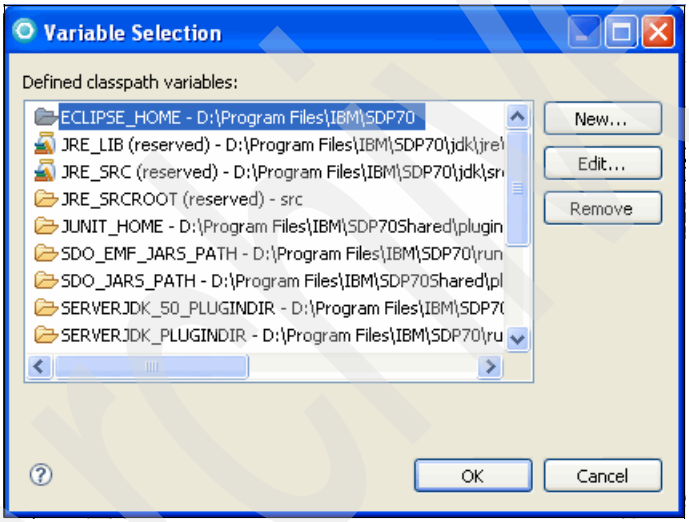


Figure 9-9 Variable selection

Archive

5. In the Edit Variable Entry window (Figure 9-10), for Name of the new variable, type `WAS_61_PLUGINDIR`. For Path, specify the directory that contains the com.ibm.events.clients_6.1.0.jar. In our sample, we type `D:\Program Files\IBM\SDP70\runtimes\base_v61\plugins`. Click **OK**.

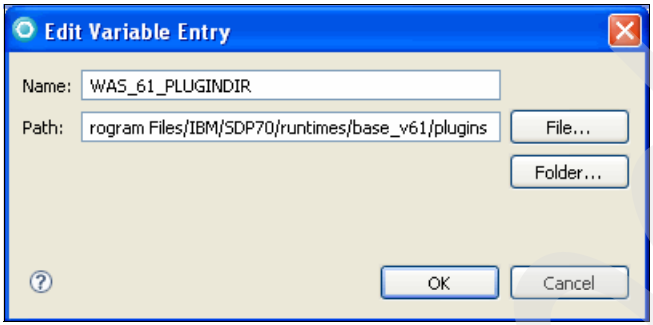


Figure 9-10 Edit Variable Entry window

6. When prompted by the Classpath Variables Changed window (Figure 9-11), click **Yes** for a full rebuild.

Figure 9-11 Classpath Variables Changed window

There are no further error messages in the Problems view.

Modifying the sample application

To modify the sample application:

1. After importing the project into the workspace, in the Project Explorer view, expand **FileEmitterEJB** → **Deployment Descriptor** and open the **FileEmitterEJB Deployment Descriptor** project.
2. In the Deployment Descriptor, click the **Source** tab.
3. A shown in Figure 9-12 on page 615, notice the values for the inbound, archive, and fault directories. Replace the default values with the inbound, archive, and fault directories that you created earlier.

Archived

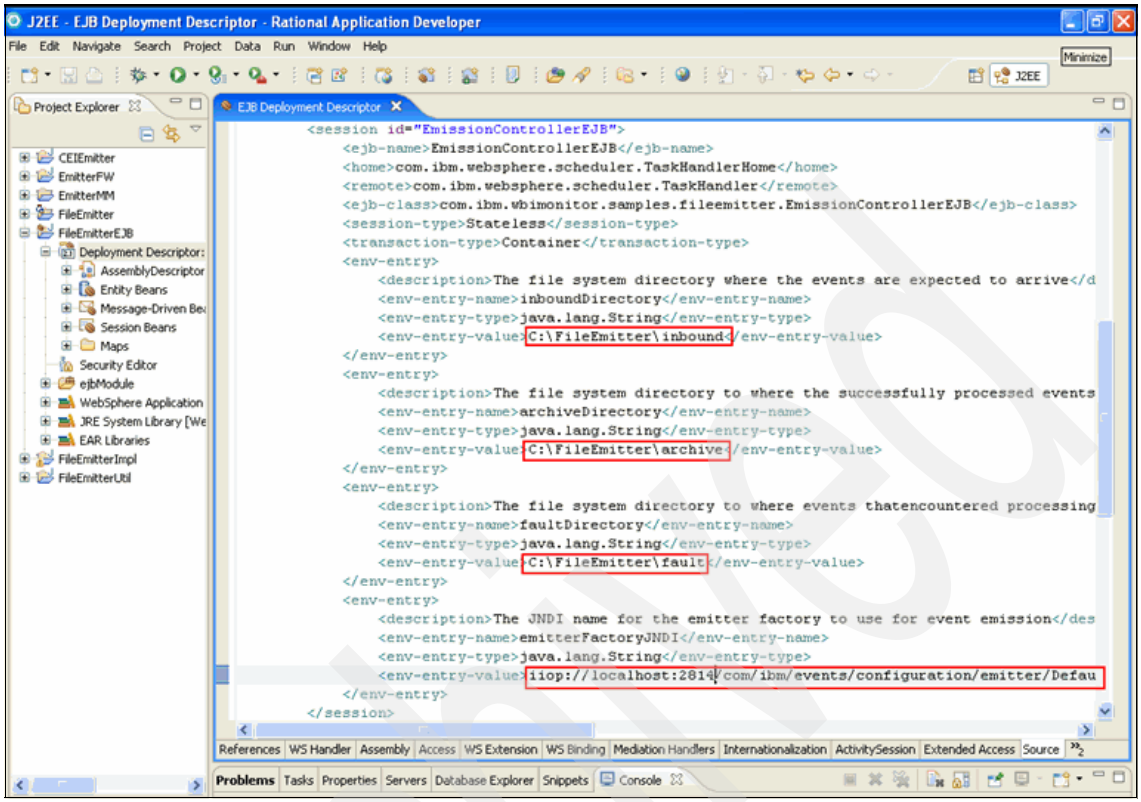


Figure 9-12 Modifying the FileEmitter EJB deployment descriptor

Configuring the emitter factory JNDI setting

The default emitter factory specified in the emitterFactoryJNDI environment entry refers to *localhost* because the emitter is deployed to the same application server as the Monitor server. This entry must refer to the *host name* of the CEI server to where the events will be emitted. If the emitter is deployed to its own application server, ensure that the host name is that of the Monitor server.

The default emitter factory specified in the emitterFactoryJNDI environment entry uses the port number 2809. Change the port number of the BOOTSTRAP_ADDRESS to that of the CEI server. If you are unsure, you can determine the port number from the WebSphere administrative console.

1. In the Servers view, right-click **WebSphere Business Monitor Server v6.1 on WebSphere Application Server** and select **Start**. Wait for the server status in the Servers view to indicate that the server has started.
2. Right-click **WebSphere Business Monitor Server v6.1 on WebSphere Application Server** again and select **Run Administrative Console**.

3. In the Integrated Solutions Console, expand **Servers** and click **Application servers**. In the right pane, click **server1**.
4. Click the **Configuration** tab.
5. Scroll down to the Communications section (Figure 9-13). Expand **Ports** and note the port number for the BOOTSTRAP_ADDRESS. In our example, it is 2814.

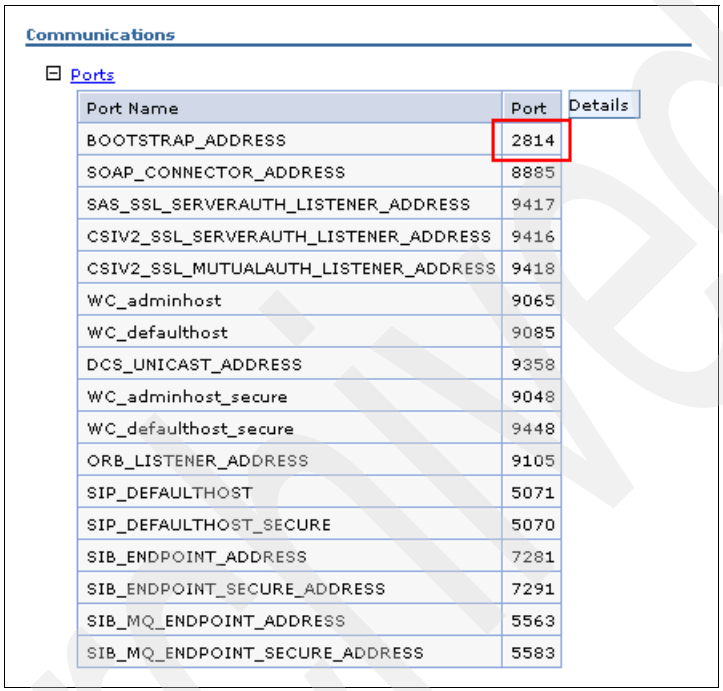


Figure 9-13 Determining the BOOTSTRAP_ADDRESS port number

6. Stop the server.
7. Modify the deployment descriptor emitterFactoryJNDI environment entry. Replace the port number with the target server's BOOTSTRAP_ADDRESS port number.
8. Save and close the deployment descriptor.

9.5.2 Creating the shipment notification event formatter

In this section, we create a custom shipment notification event formatter for converting the data from the shipping notification file to a Common Base Event formatted event.

Archived

Creating the shipment notification XML schema

To create the shipment notification XML schema:

1. In Rational Application Developer, under the Project Explorer view (Figure 9-14), right-click the **FileEmitterImpl** Java Project and select **New** → **Other**.

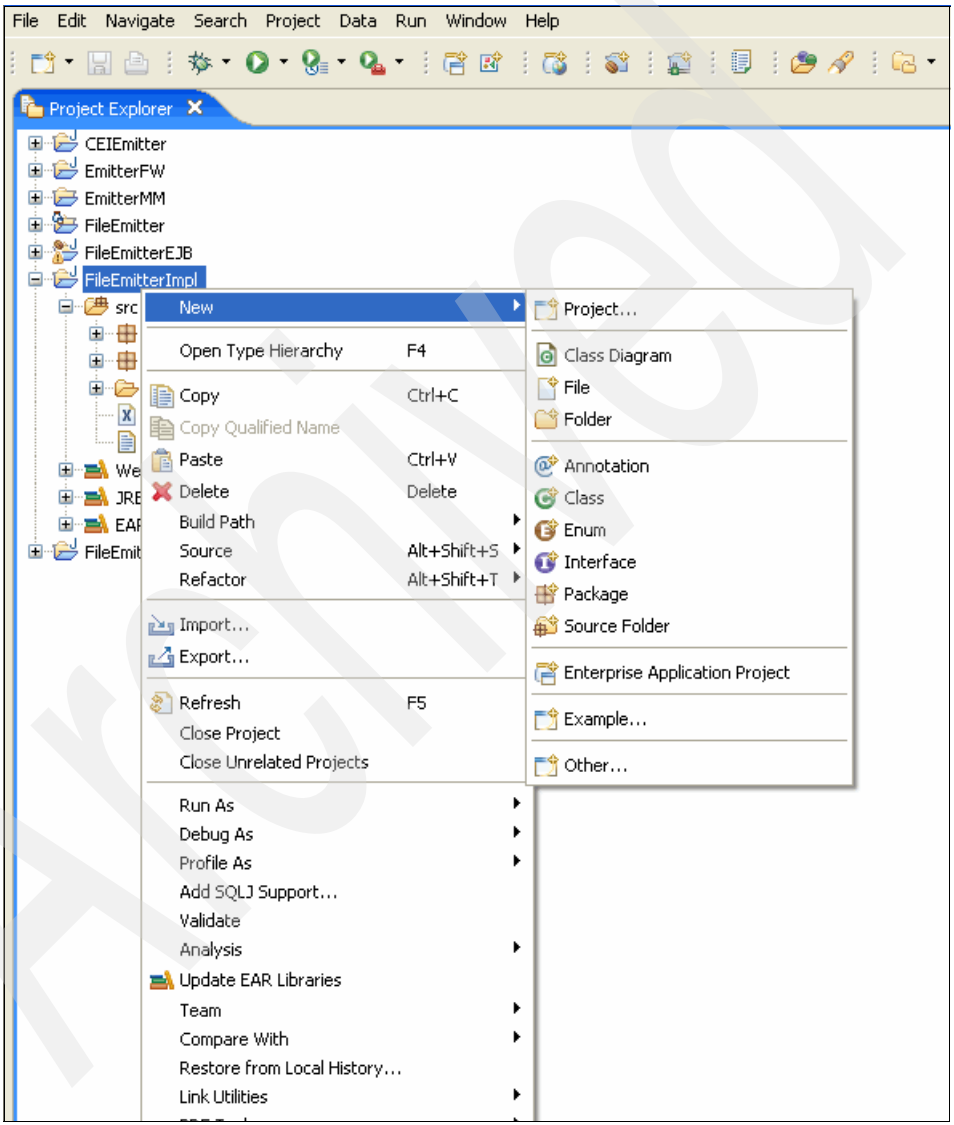


Figure 9-14 Adding a new XML schema

Archive

2. In the New – Select a wizard window (Figure 9-15), expand **XML** and select **XML Schema**. Click **Next**.

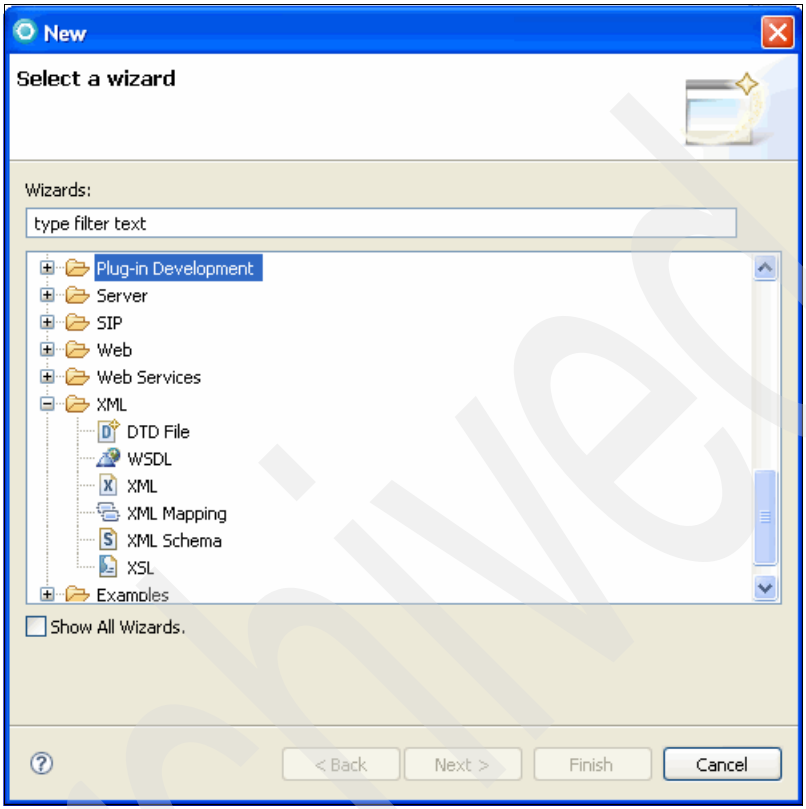


Figure 9-15 XML Schema wizard window

Archived

3. In the Create XML Schema – New XML Schema window (Figure 9-16), expand **FileEmitterImpl** and select **src**. For File name of the XML Schema, type `ShipmentNotification.xsd`. Click **Finish**.

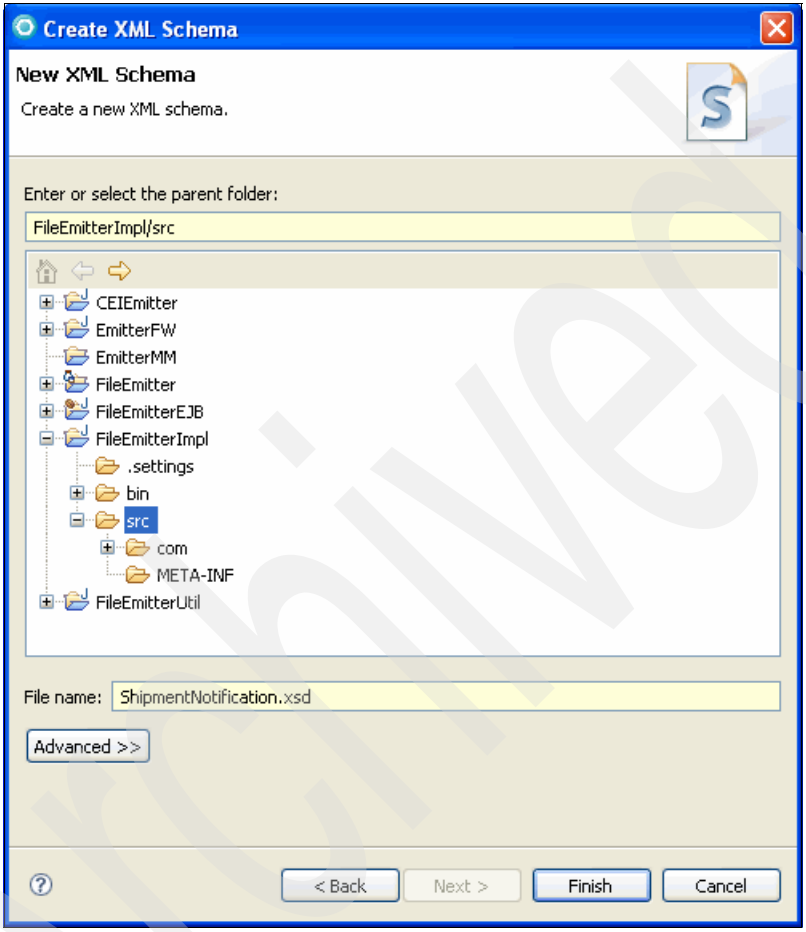


Figure 9-16 Specifying the target location of the XML schema

Archived

4. In the newly created XML schema (Figure 9-17), right-click the **Types** pane and select **Add Complex Type**.

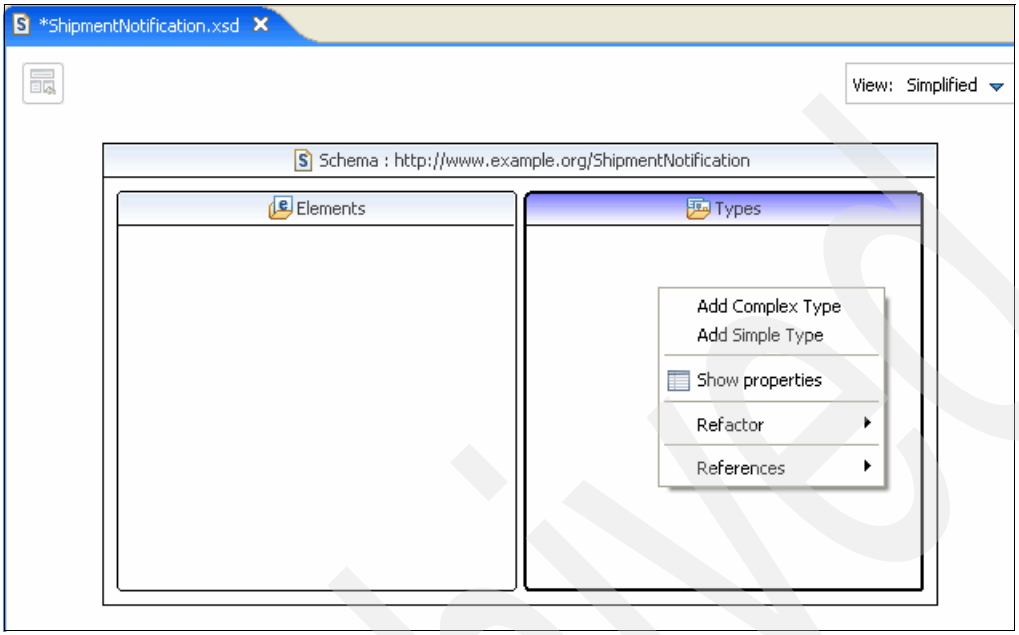


Figure 9-17 ShipmentNotification XML schema

5. Type `ShipmentNotification` as the name of the new complex type (Figure 9-18).

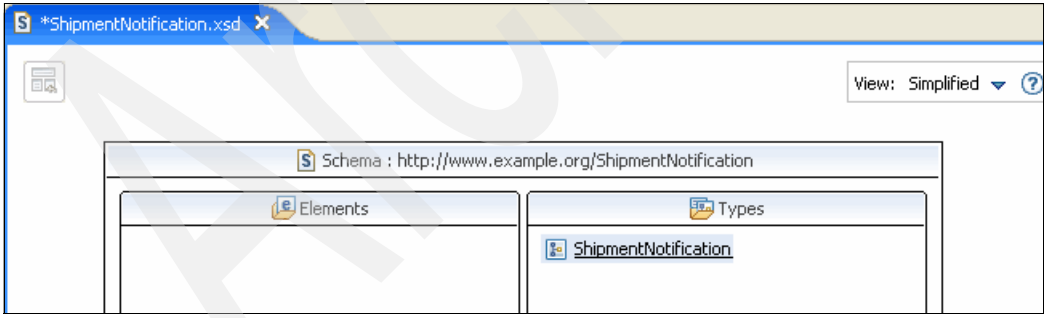


Figure 9-18 Shipment Notification type

Archived

6. As shown in Figure 9-19, double-click the newly created **Shipment Notification** type. Right-click **ShipmentNotification** and select **Add Element**.

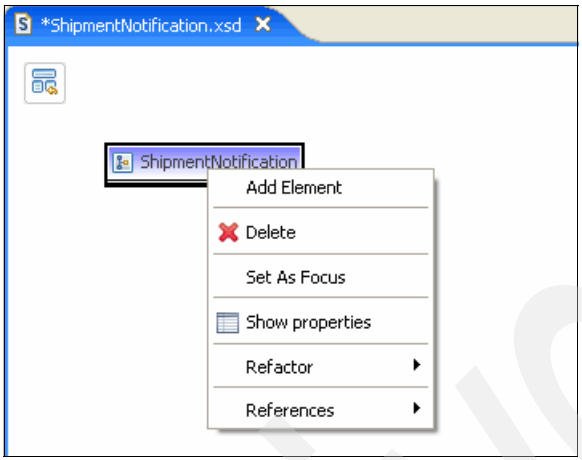


Figure 9-19 Adding elements

7. Add the elements as shown in Figure 9-20. When finished, click the icon in the left corner of the page (highlighted in Figure 9-20).

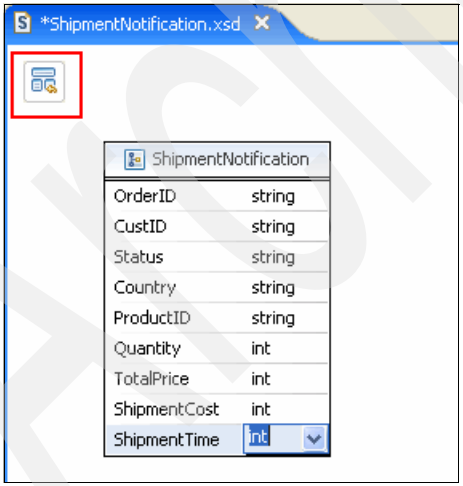


Figure 9-20 Elements in Shipment Notification

Archived

8. Right-click the **Elements** tab and select **Add Element** (Figure 9-21).

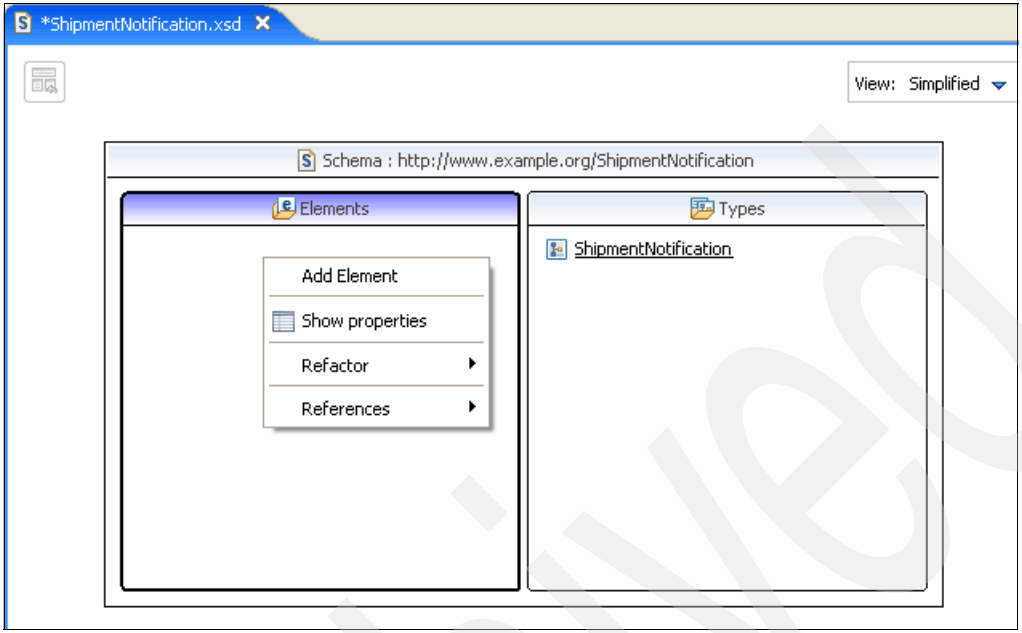


Figure 9-21 Adding elements

9. Type `ShipmentNotification` as the name of the element. Right-click the name and select **Set Type** → **Browse**. Select **ShipmentNotification** as the type and click **OK**.

Archived

Figure 9-22 shows the completed XML Schema Definition (XSD). Note the namespace of the schema, `http://www.example.org/ShipmentNotification`, which you will use later in this chapter.

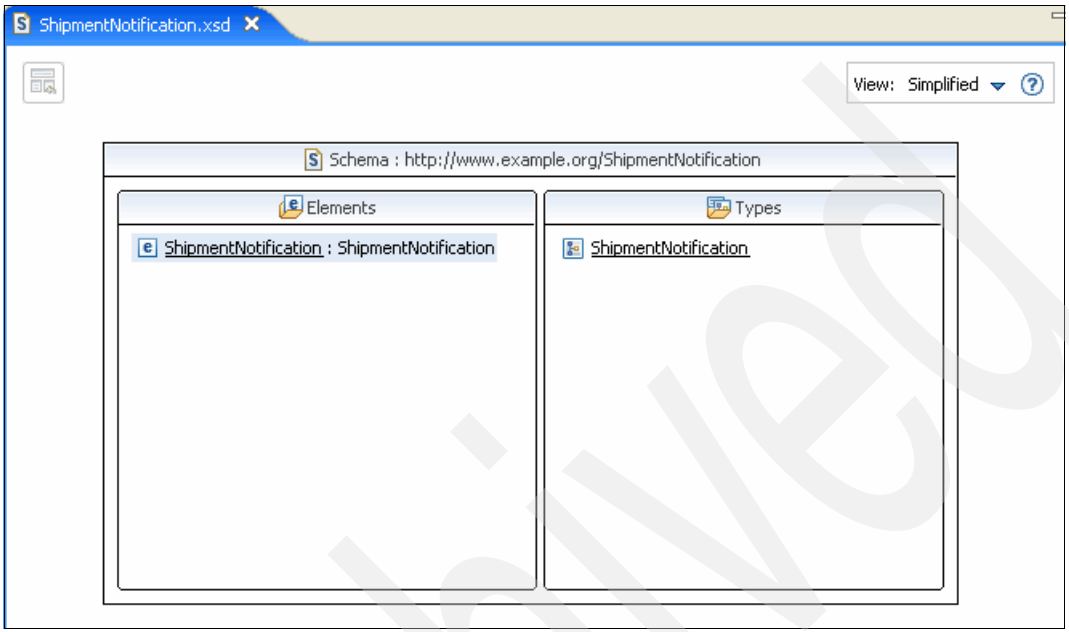


Figure 9-22 Completed ShipmentNotification.xsd

Alternatively, instead of using the XML schema wizard, you can create an XML schema as shown in Example 9-10.

Example 9-10 ShipmentNotification.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.example.org/ShipmentNotification"
xmlns:tns="http://www.example.org/ShipmentNotification"
elementFormDefault="qualified">

   <complexType name="ShipmentNotification">
   <sequence>
      <element name="OrderID" type="string"></element>
      <element name="CustID" type="string"></element>
      <element name="Status" type="string"></element>
      <element name="Country" type="string"></element>
      <element name="ProductID" type="string"></element>
      <element name="Quantity" type="int"></element>
```

Archived

```
      <element name="TotalPrice" type="int"></element>
      <element name="ShipmentCost" type="int"></element>
      <element name="ShipmentTime" type="int"></element>
   </sequence>
    </complexType>

    <element name="ShipmentNotification"
    type="tns:ShipmentNotification">
    </element>
</schema>
```

Best practices: When creating XSD files, you can make it easier to use the files for event definitions in the Monitor Model editor. To do this, follow the best practices in the "Configuring a remote CEI server to use WebSphere Business Monitor" topic in the WebSphere Business Monitor information center at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.btools.help.monitor61.doc/admin/cei_cfg_for_rem_monitor.html

Creating the shipment notification formatter

In this section, we create a shipment notification formatter to format application data to a Common Base Event. We provide the skeleton class that you can build from. Subsequently, you can use the completed formatter as a reference for building your own formatter for other application data.

To create the shipment notification formatter:

1. The skeleton file is available in the project hierarchy. In the Project Explorer view, expand **FileEmitterImpl** → **src** → **com.ibm.wbimonitor.samples.fileemitter.formatter.xsd** → **ShipmentNotificationFormatterImpl.java**.
2. In the definition of the schema (Example 9-11):
   a. Initialize the variable SCHEMA to the XML schema definition that we created earlier. In our example, this is `"/ShipmentNotification.xsd"`.
   b. Similarly, initialize the variable NAMESPACE to the namespace of the schema created earlier. In our example, this is `"http://www.example.org/ShipmentNotification"`.

Example 9-11 Shipment notification schema and namespace definition

```
private static final String SCHEMA = "/ShipmentNotification.xsd";
private static final String NAMESPACE =
"http://www.example.org/ShipmentNotification";
```

3. In ShipmentNotificationFormatterImpl.java, scroll down to the line of code with the statement "`Insert code snippet here`." Copy and paste the text from /eventemitter/Shipment Notification Code Snippet.txt after it as shown in Example 9-12. This code fragment creates the XML document per the XML schema, ShipmentNotification.xsd, that we defined earlier.

Example 9-12 Inserting the create Common Base Event code snippet

```
//Insert code snippet here
Element shipment = document.createElementNS( NAMESPACE, "tns:ShipmentNotification" );
Element element = document.createElementNS( NAMESPACE, "tns:TRIGGER" );
element.appendChild( document.createTextNode( eventAction ) );
shipment.appendChild( element );

element = document.createElementNS( NAMESPACE, "tns:OrderID" );
element.appendChild( document.createTextNode( eventData.getProperty("OrderID")));
shipment.appendChild( element );

if ( !"DELETE".equalsIgnoreCase( eventAction ) )
{
element = document.createElementNS( NAMESPACE, "tns:CustID" );
element.appendChild( document.createTextNode( eventData.getProperty("CustID")));
shipment.appendChild( element );

element = document.createElementNS( NAMESPACE, "tns:Status" );
element.appendChild( document.createTextNode( eventData.getProperty("Status")));
shipment.appendChild( element );

element = document.createElementNS( NAMESPACE, "tns:Country" );
element.appendChild( document.createTextNode( eventData.getProperty("Country")));
shipment.appendChild( element );

element = document.createElementNS( NAMESPACE, "tns:ProductID" );
element.appendChild( document.createTextNode( eventData.getProperty("ProductID")));
shipment.appendChild( element );

element = document.createElementNS( NAMESPACE, "tns:Quantity" );
element.appendChild( document.createTextNode( eventData.getProperty("Quantity")));
shipment.appendChild( element );

element = document.createElementNS( NAMESPACE, "tns:TotalPrice" );
element.appendChild( document.createTextNode( eventData.getProperty("TotalPrice")));
shipment.appendChild( element );

element = document.createElementNS( NAMESPACE, "tns:ShipmentCost" );
```

Archived

```
element.appendChild( document.createTextNode(eventData.getProperty("ShipmentCost")));
shipment.appendChild( element );

element = document.createElementNS( NAMESPACE, "tns:ShipmentTime" );
element.appendChild( document.createTextNode(eventData.getProperty("ShipmentTime")));
shipment.appendChild( element );
}
document.appendChild( shipment );

//end insert code snippet
```

4. Save your work.

Event formats: The file emitter sample includes event formatters that create events in the WebSphere Business Monitor v6.0.2 format (earlier XML) and the WebSphere Business Monitor v6.1 format (XML with schema support). In our scenario, the shipment notification events are generated in the WebSphere Business Monitor v6.1 format.

In the package, com.ibm.wbimonitor.samples.fileemitter.formatter, you see event formatters that format events in the WebSphere Business Monitor v6.0.2 format.

Modifying the properties file

After creating the shipment notification event formatter, we update the FileEventFormatter.properties file to reflect the new event formatter. Files that are tagged with a type of SHIPMENT are processed by this new event formatter. To modify the properties file:

1. In the Project Explorer, expand **FileEmitterImpl** → **src** → **FileEventFormatter.properties**.
2. In the properties file (Example 9-13), add the following line:

```
SHIPMENT=com.ibm.wbimonitor.samples.fileemitter.formatter.xsd.Shipme
ntNotificationFormatterImpl
```

Example 9-13 FileEventFormatter.properties

```
#
# This properties file defines the formatter to use for a given data type.
# The accepted format is:
#     EventType=EventFormatter
# Where the EventFormatter is provided as the fully qualified Java class.
#
#CUSTOMER=com.ibm.wbimonitor.samples.fileemitter.formatter.CUSTOMERFormatterImpl
```

```
CUSTOMER=com.ibm.wbimonitor.samples.fileemitter.formatter.xsd.CUSTOMERFormatterImpl
ORDER=com.ibm.wbimonitor.samples.fileemitter.formatter.ORDERFormatterImpl
CLAIM=com.ibm.wbimonitor.samples.fileemitter.formatter.CLAIMFormatterImpl
SHIPMENT=com.ibm.wbimonitor.samples.fileemitter.formatter.xsd.ShipmentNotificationFor
matterImpl
```

3. Save your work.

9.6 Configuring and deploying the emitter application

In this section, we explain how to configure the WebSphere Business Monitor server to host the Sample File Event Emitter application.

9.6.1 Setting up the WebSphere Business Monitor server scheduler

In the file event emitter sample, we use the WebSphere Application Server scheduler service. The scheduler service is a WebSphere programming extension that is responsible for starting actions at specific times or intervals.

Creating a database for the scheduler

Each scheduler requires a database in which to store its persistent information. Schedulers use this database for storing tasks and then running them. To create a database for the scheduler:

1. At a command prompt, type the following command to open the DB2 Command Line Processor:

   ```
   db2cmd
   ```

2. Type the following command to create the database:

   ```
   DB2 create database SKDLR
   ```

Creating an authentication alias for the DB2 database

To create an authentication alias for the DB2 database:

1. In the Servers view, right-click **WebSphere Business Monitor Server v6.1 on WebSphere Application Server** and select **Start**. Wait for the server status in the Servers view to indicate that the server has started.
2. Right-click **WebSphere Business Monitor Server v6.1 on WebSphere Application Server** again and select **Run Administrative Console**.
3. In the Integrated Solutions Console, expand **Security** and select **Secure administration, applications, and infrastructure**.

Archived

4. Under the Authentication heading, expand **Java Authentication and Authorization Service.**
5. Select **J2C Authentication data**.
6. Click **New** to create a new authentication alias.
7. Under General Properties, fill in the properties of the authentication alias using the details from Table 9-2 and as shown in Figure 9-23. In our example, the User ID and Password are `db2admin`. After you complete the appropriate values for your environment, click **OK**.

Table 9-2 Completing the Authentication Alias properties

Properties	Value
Alias	FileEmitterAlias
User ID	db2admin
Password	db2admin
Description	Authentication Alias for the File Event Emitter sample

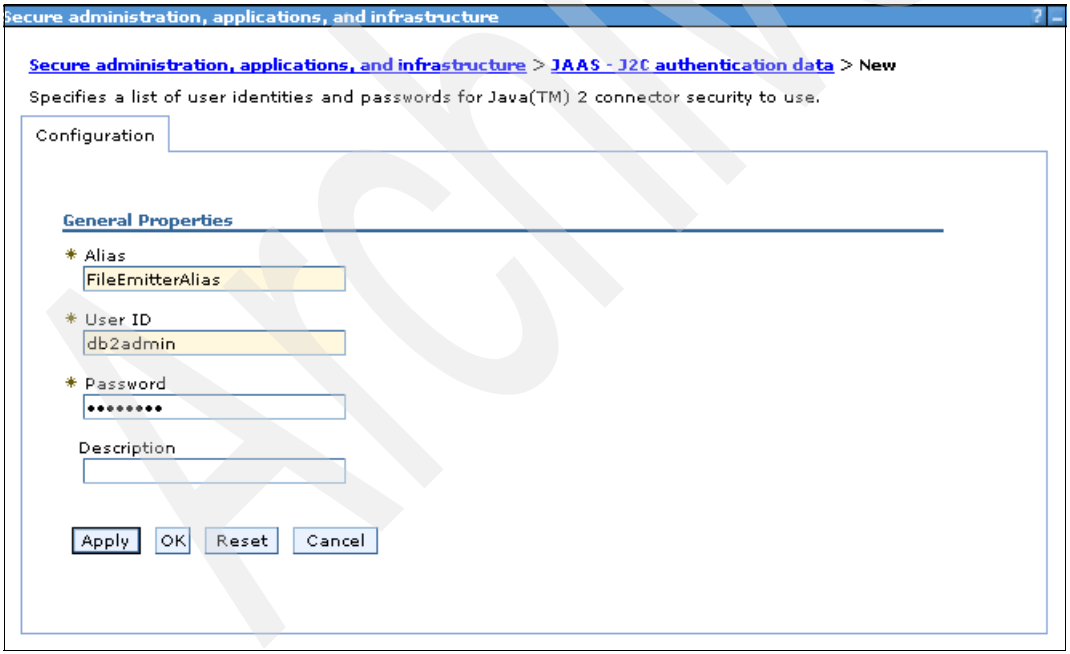


Figure 9-23 Authentication alias details

Archived

Creating a JDBC Provider for the scheduler

To create a JDBC Provider for the scheduler:

1. In the Administrative Console (Figure 9-24), expand **Resources** → **JDBC** and click **JDBC Providers**. In the right pane, set the scope to **server**. Click the **New** button to create a new JDBC provider.

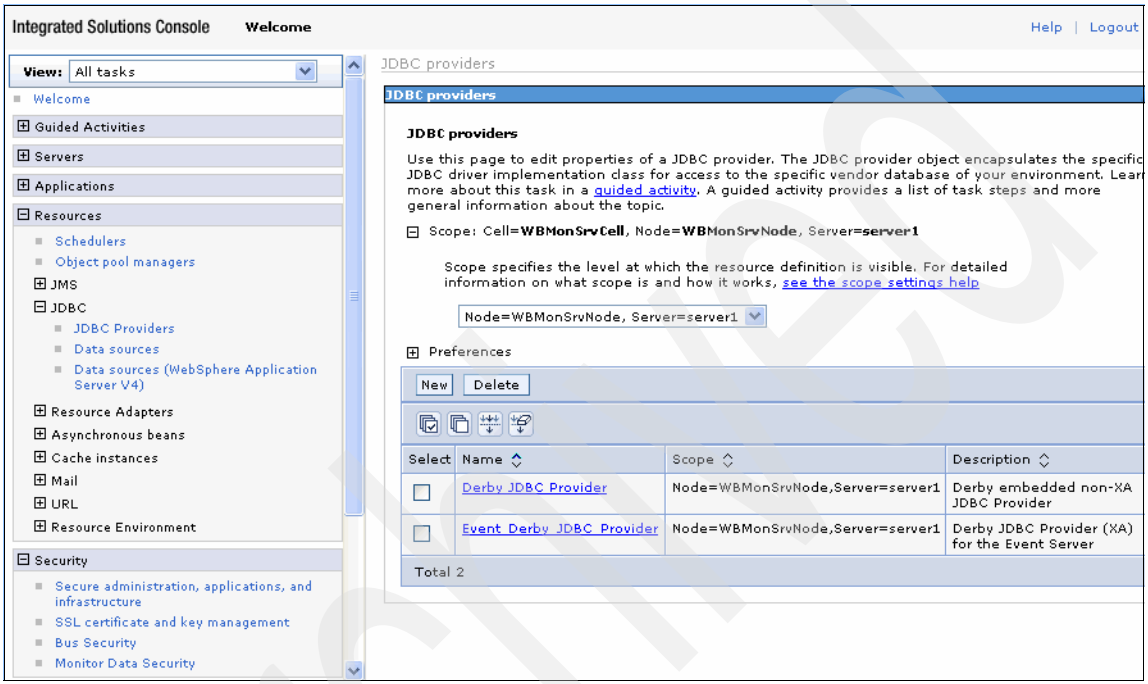


Figure 9-24 Setting the scope

Archived

2. In the Create a new JDBC Provider panel (Figure 9-25), set the properties as provided in Table 9-3.

Table 9-3 Basic configuration properties of JDBC provider

Properties	Values
Database Type	DB2
Provider Type	DB2 Universal JDBC Driver Provider
Implementation Type	XA data source

When complete, click **Next**.

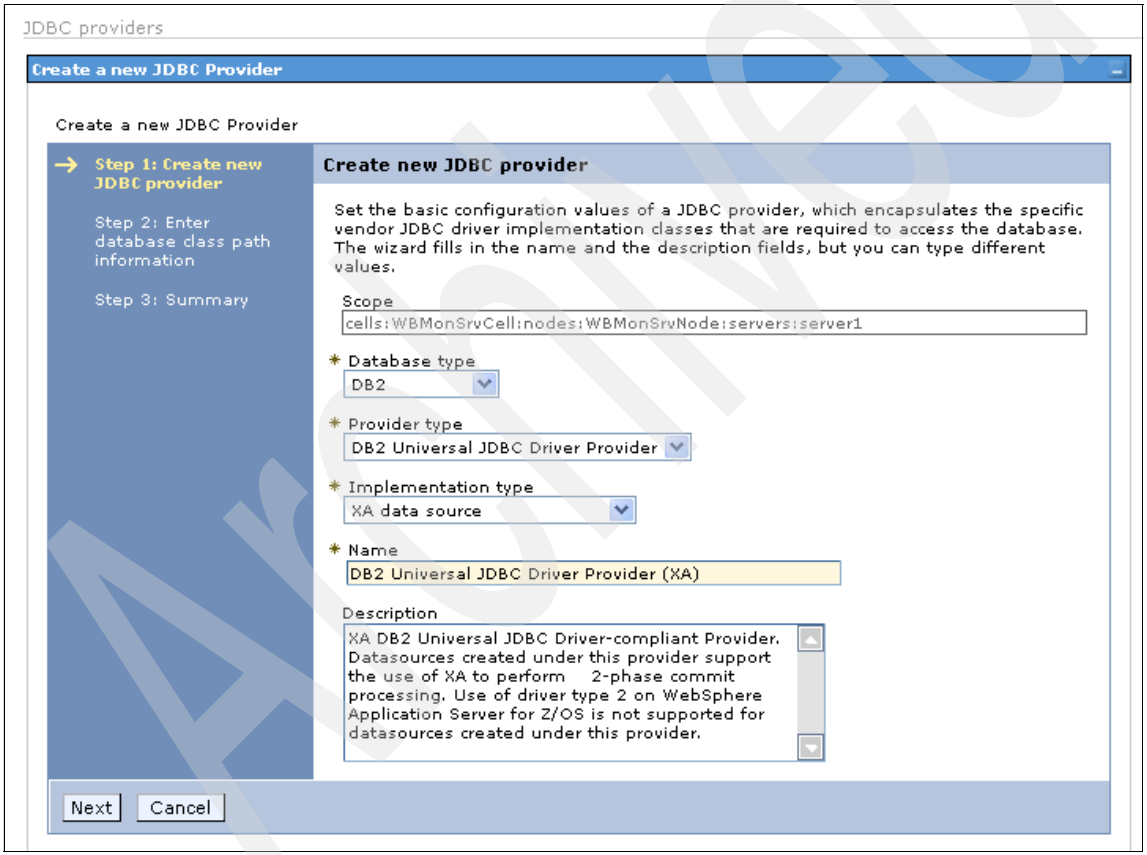


Figure 9-25 Create new JDBC provider panel

Archived

3. In the Enter database class path information panel (Figure 9-26), specify the necessary information. In our example, we type `D:\Program Files\IBM\SQLLIB\java` for ${DB2UNIVERSAL_JDBC_DRIVER_PATH} and ${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}. This is the location where the DB2 Java libraries are located. Click **Next**.

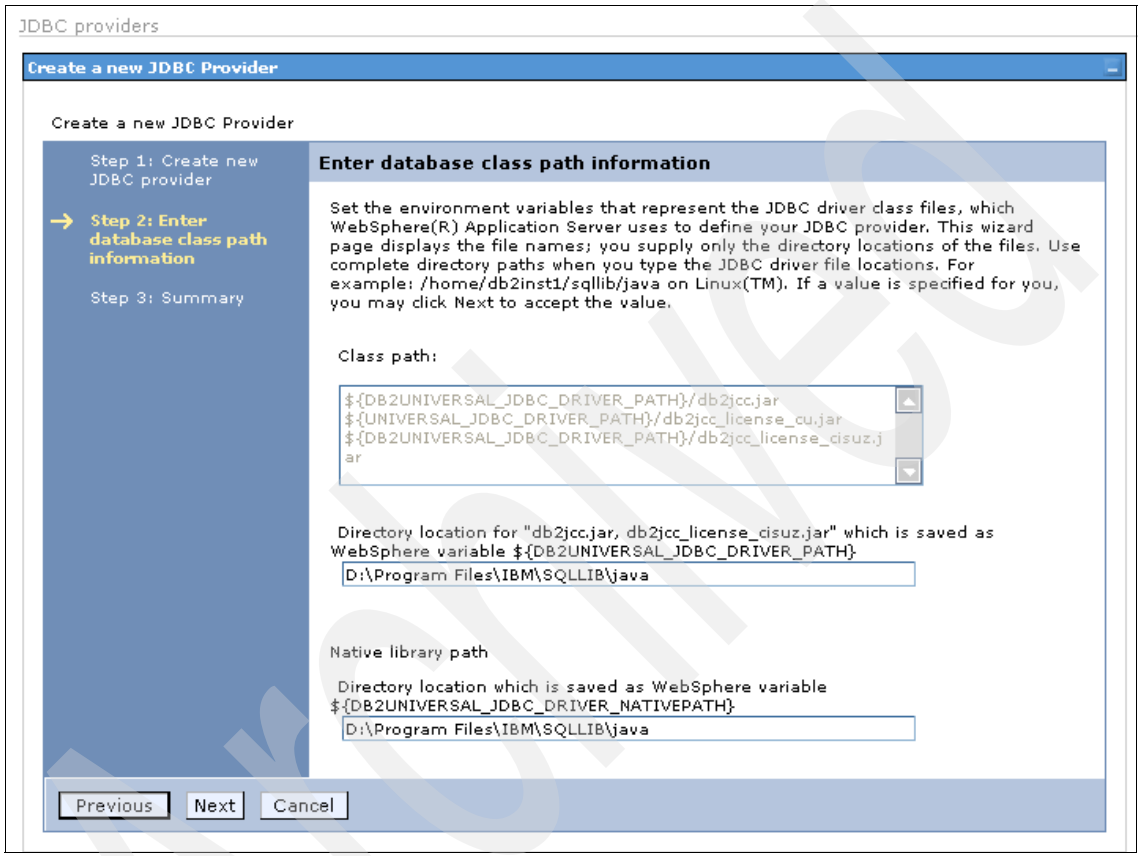


Figure 9-26 Entering the database classpath information

4. Review the Summary of actions and click **Finish**.
5. Click the **Save** directly to the master configuration link.

Your new JDBC Provider is now listed as shown in Figure 9-27.

Figure 9-27 DB2 JDBC Provider created

Archived

Creating a data source for the scheduler

To create a data source for the scheduler:

1. In the JDBC providers panel (Figure 9-28), click the newly created **DB2 Universal JDBC Provider (XA)** and then click the **Data sources** link.

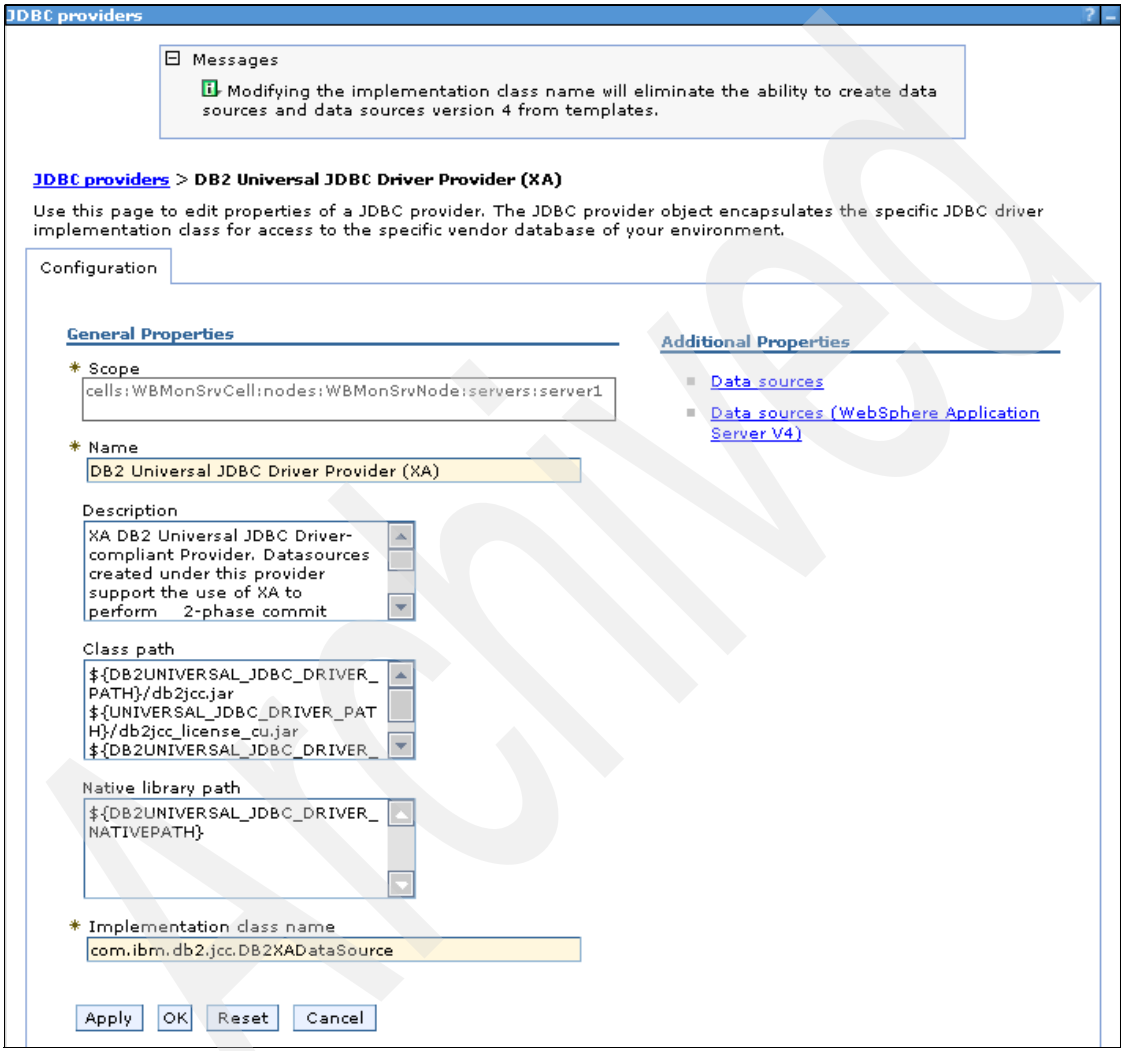


Figure 9-28 Newly created JDBC provider

Archived

2. in the Data sources panel (Figure 9-29), click **New** to create a new data source.

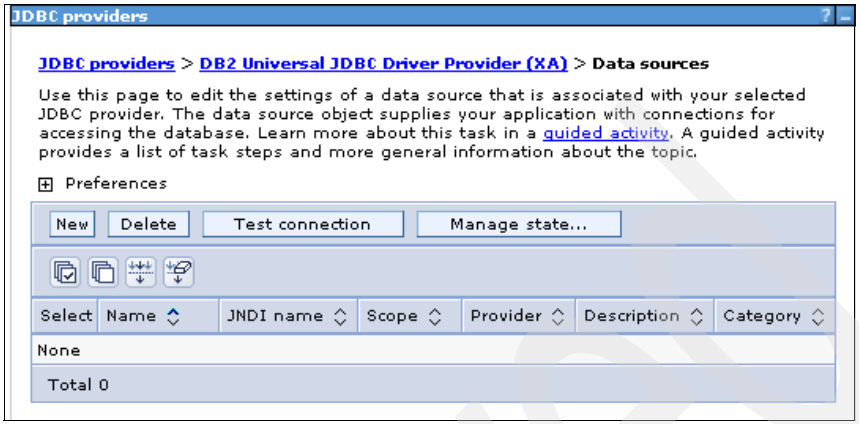


Figure 9-29 Creating a new data source

Archive

3. In the Enter basic data source information panel (Figure 9-30), enter the values from Table 9-4 and click **Next**.

Table 9-4 Properties of the basic data source information

Properties	Values
Name	SKDLR
JNDI name	jdbc/skdlr
Component-managed authentication alias	FileEmitterAlias

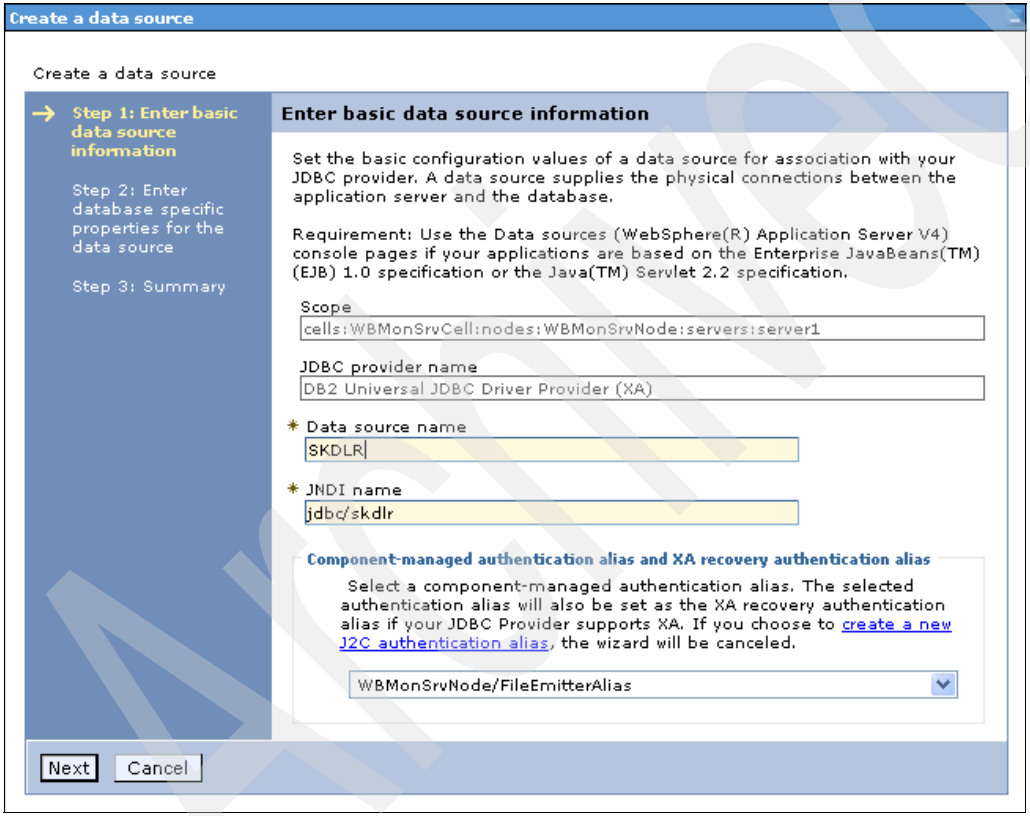


Figure 9-30 Entering the basic data source information

Archived

4. In the Enter database specific properties for the data source panel (Figure 9-31), enter the values from Table 9-5 and click **Next**.

Table 9-5 Properties of the data source database

Properties	Values
Database name	SKDLR
Server name	Host name of DB2 server machine
Port number	50000 (default)

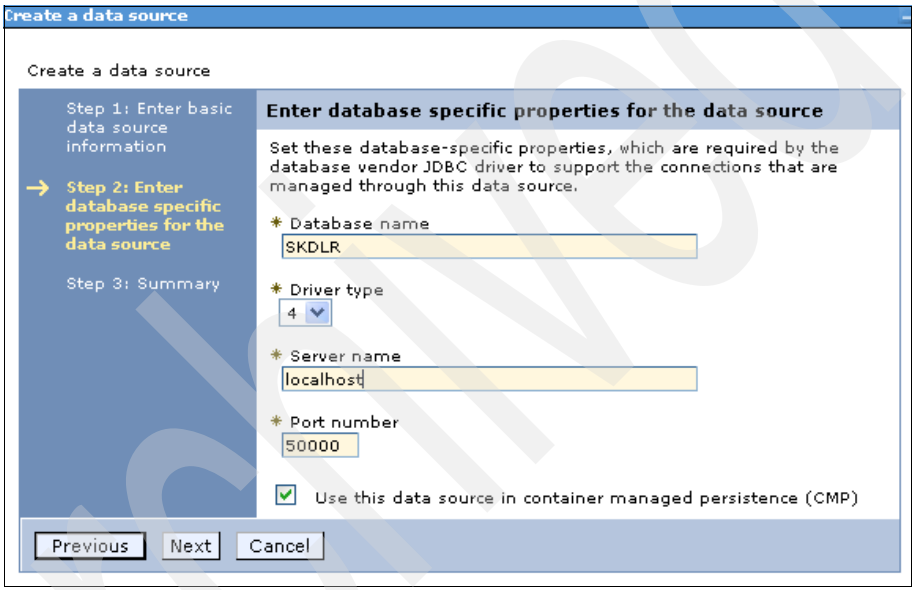


Figure 9-31 Entering the database specific properties for the data source

5. Review the summary of actions and click **Finish**.
6. Click the **Save** directly to the master configuration link.
7. Click the **Test Connection** button to test the connectivity to the data source.

Archived

Creating a scheduler for the event emitter

The scheduler is required to initiate the task of polling the file directories at regular intervals for the inbound shipment notification file. To create a scheduler:

1. In the Integrated Solutions Console (Figure 9-32), select **Resources** → **Scheduler**. In the right pane, set scope to **server**. Click **New** to create a new scheduler.

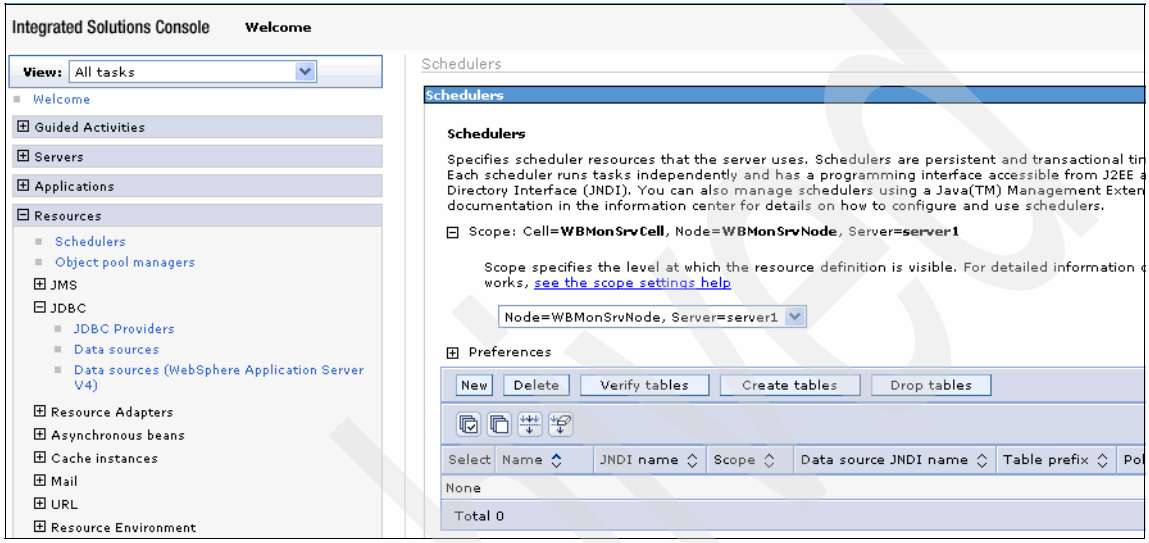


Figure 9-32 Setting the scheduler scope

2. Under General Properties of the new scheduler (Figure 9-33 on page 638), enter the values from Table 9-6. Then click **OK**.

Table 9-6 Scheduler values for File Event Emitter

Properties	Values
Name	FolderPoller
JNDI name	sched/Poller (This must be the same as the environment entry in the EJB deployment descriptor.)
Description	Scheduler for File Sample Emitter's Inbound Folder Poller
Data Source JNDI name	jdbc/skdlr
Table prefix	FILEEMTR_
Poll interval	30
Work managers	wm/default

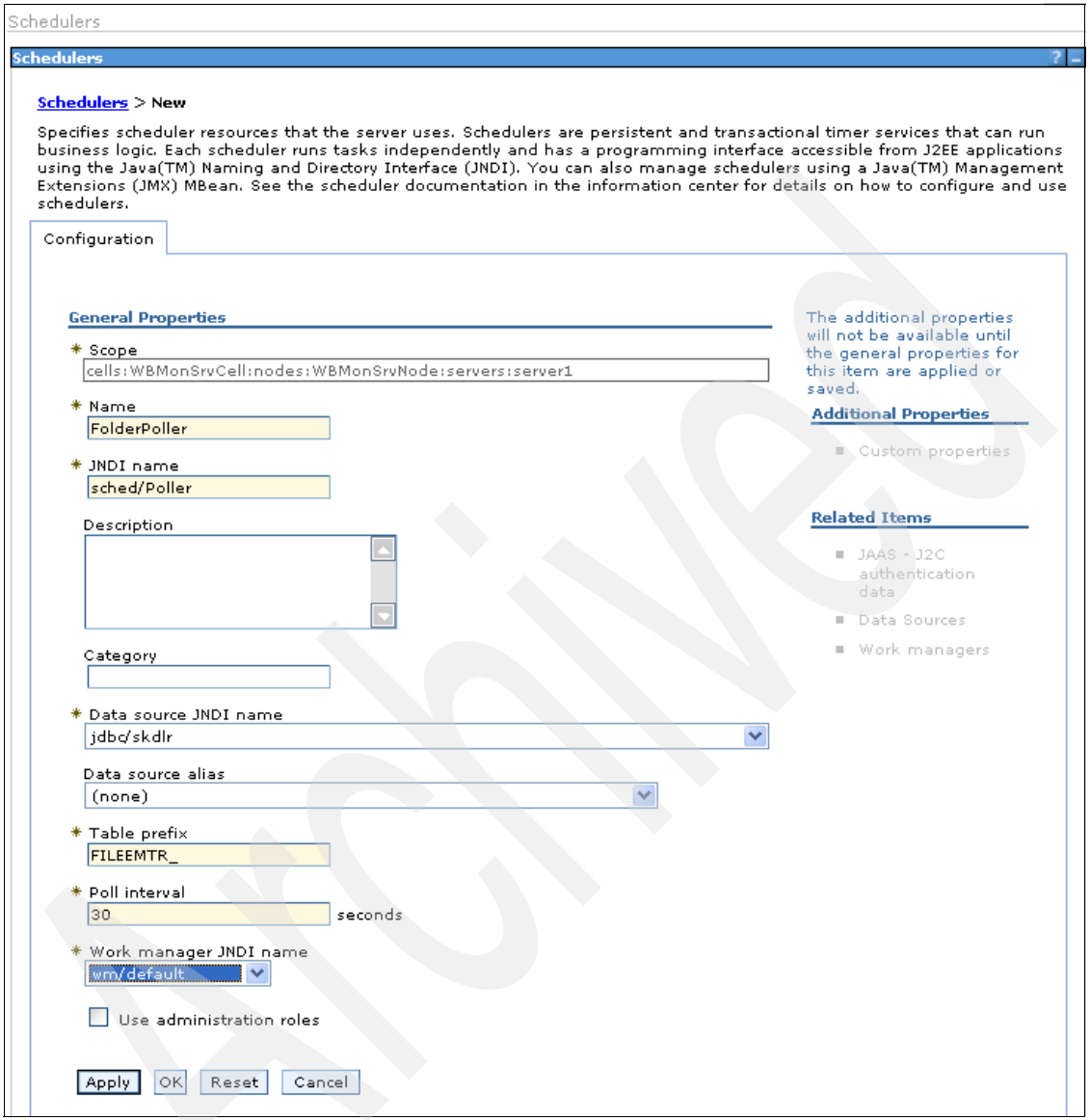


Figure 9-33 Configuring the scheduler properties

3. Click the **Save** directly to the master configuration link.

Archived

4. In the Scheduler panel (Figure 9-34), select the newly created **FolderPoller** scheduler and click **Create tables**.

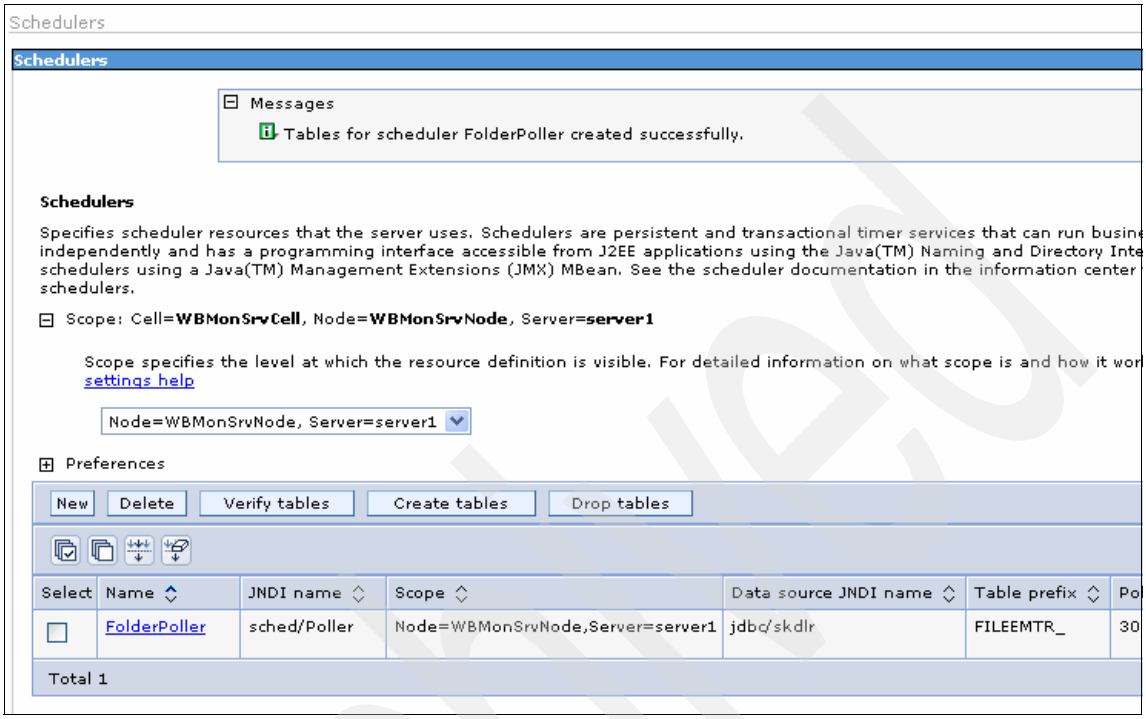


Figure 9-34 Creating tables for the scheduler

If successful, you see a message that indicates that the tables for scheduler FolderPoller are created successfully (Figure 9-35).

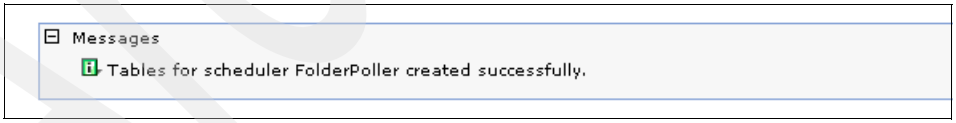


Figure 9-35 Table for scheduler FolderPoller created successfully

5. Restart the server to run the scheduler tasks.

Archived

9.6.2 Deploying the sample event emitter application

After configuring the resources on the WebSphere Application Server, we deploy the sample event emitter application to the integrated test environment server in Rational Application Developer:

1. In the Rational Application Developer, from the Servers view, right-click **WebSphere Business Monitor v6.1 on WebSphere Application Server** and select **Add and Remove Project**.
2. In the Add and Remove Projects window (Figure 9-36), add the FileEmitterEAR to the server (Configured projects) and click **Finish**.

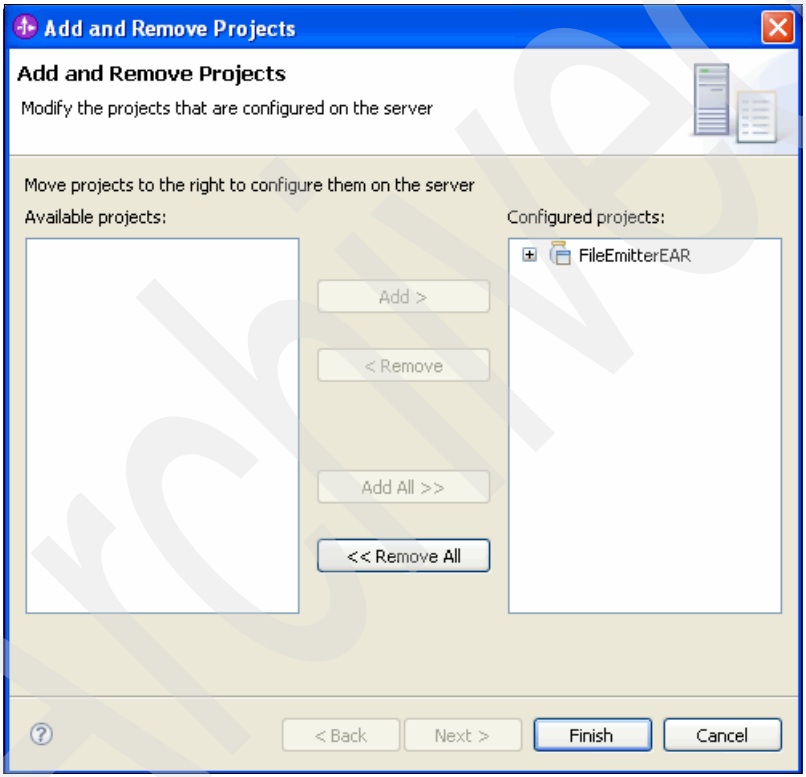


Figure 9-36 Adding FileEmitterEAR to the Monitor server

Troubleshooting

The application deploys successfully. Some of the common error messages are inconsistent definition of resource names and missing configuration information. For example, check the SCHEMA and NAMESPACE definitions in ShipmentNotificationFormatterImpl.java.

Archived

Retrying the installation

To retry the installation of the application:

1. Uninstall the application from the Integrated Solutions Console.
2. Stop the server.
3. Start the server.
4. Add the projects again.

9.6.3 Testing the emitter

To test the emitter:

1. Navigate to the inbound directory created earlier. In our example, this directory is C:\FileEmitter\inbound. Copy the file /eventemitter/Sample/Sample.txt to the directory. This file is provided with the additional materials for this book. For more information, see Appendix B, “Additional material” on page 801.
2. If successful, the file is no longer displayed and a copy is in the archive directory. Behind the scenes, the scheduler service has initiated the FileEmitter application to retrieve the file from the inbound directory and converts it to the Common Base Event format for emission to WebSphere Business Monitor. It also places a copy of the original file in the archive folder for archiving purposes.
3. Run the **eventquery** script in the bin directory of the Monitor server. In our example, the directory is D:\Program Files\IBM\SDP70\runtimes\base_v61\profiles\WBMonSrv\bin. To run the script, enter the following command:

   ```
   eventquery.bat -group "All events"
   ```

   Figure 9-37 on page 642 shows the output from running the script.

Archived

Figure 9-37 Eventquery.bat

If successful, a corresponding event is returned with details of the Common Base Event displayed.

9.7 Creating the monitor model

After testing the event is successfully emitted, we create a corresponding monitor model to be deployed on the Monitor server.

Creating a business monitoring project

To create a business monitoring project:

1. In Rational Application Developer, select **File** → **New** → **Project**.
2. In the New Project – Select a wizard window (Figure 9-38 on page 643), expand **Business Monitoring** and select **Business Monitoring Project**. Click **Next**.

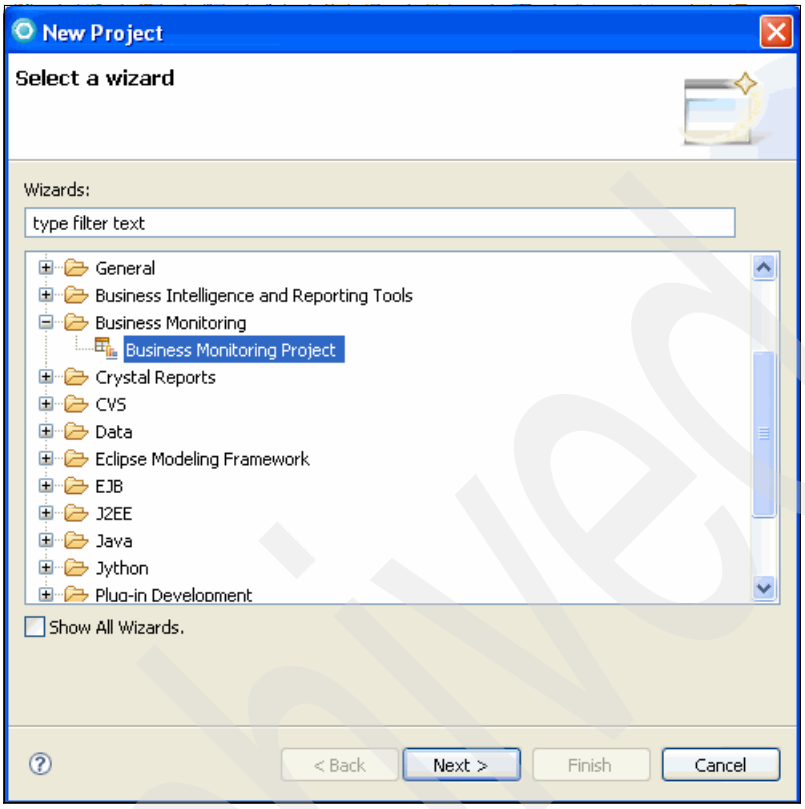


Figure 9-38 Creating a new Business Monitoring Project

3. In the New Business Monitoring Project window (Figure 9-39), for Project name, type `ShipmentNotificationBM` and click **Finish**.

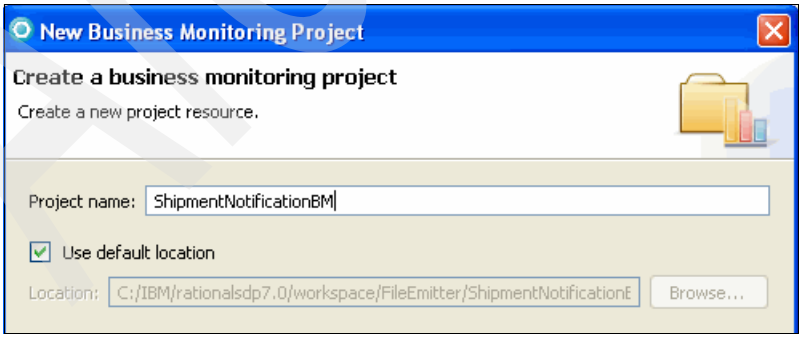


Figure 9-39 ShipmentNotificationBM Business Monitoring Project

4. Select **Yes** to open the Business Monitoring perspective and select **No** to launch Get Started.
5. In the Project Explorer, right-click the newly created **ShipmentNotificationBM** and select **Associate Projects**.
6. In the Project References window (Figure 9-40), select the **FileEmitterImpl** project to reference it. We reference the FileEmitterImpl because the XML schema, ShipmentNotification.xsd, was defined in the project. Click **OK**.

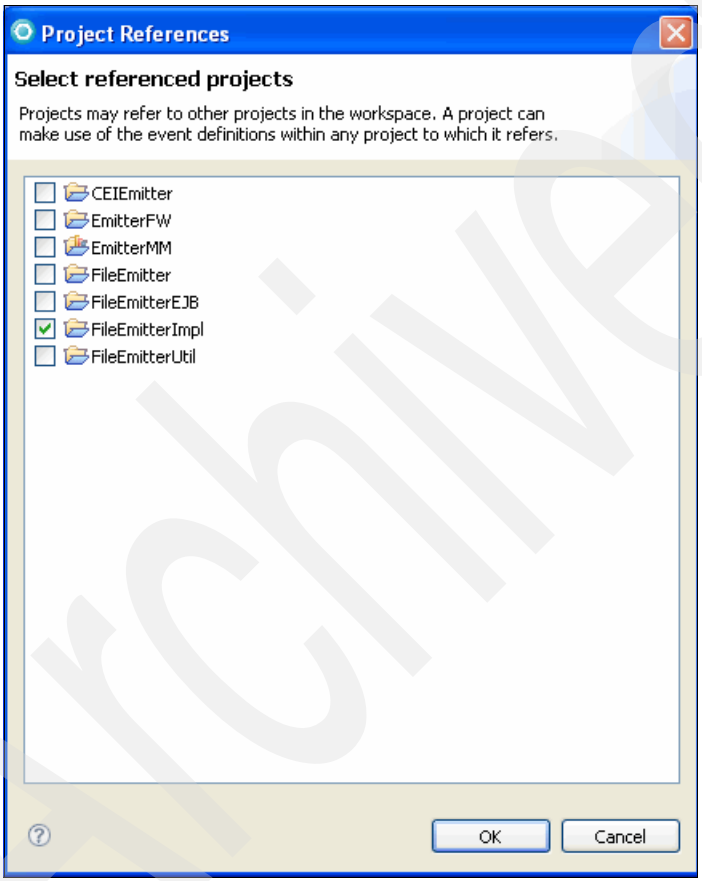


Figure 9-40 Referencing FileEmitterImpl

Archived

The ShipmentNotification.xsd event definition has been created as shown in Figure 9-41.

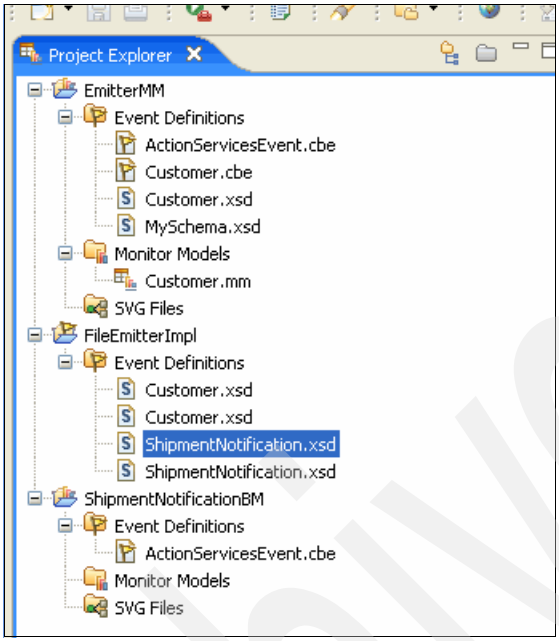


Figure 9-41 ShipmentNotification.xsd

Creating the monitor model

To create the monitor model:

1. In the Project Explorer (Figure 9-42), right-click **ShipmentNotificationBM** and select **New** → **Monitor Model**.

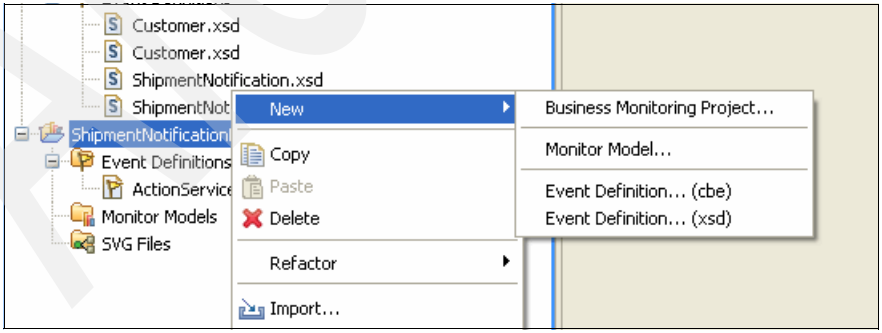


Figure 9-42 Creating a new monitor model

Archived

2. When prompted, select **ShipmentNotificationBM** as the parent folder and type `ShipmentNotificationMM` as the file name. Click **Finish**.
3. Click **No** if prompted to launch Get Started information.

Creating an inbound event definition

A monitoring-enabled application generates a series of events. To indicate the events that are of interest to the monitoring context, you define inbound events in the Monitor Model editor. These are subscription points or entry points for inbound events in a monitoring context. WebSphere Business Monitor subscribes to the events that you specify and delivers them to all event entry points with a matching event subscription.

Because the events are often generated elsewhere and WebSphere Business Monitor simply monitors them, you usually import the definitions of the inbound events from somewhere else.

To create an inbound event definition:

1. In the Monitor Details Model, right-click **ShipmentNotificationMM MC** and select **New** → **Inbound Event**.
2. For the name of the inbound event, type `ShipmentEvent` and click **OK**.
3. In the Monitor Details Model panel (Figure 9-43), in the left pane, select the newly created **ShipmentEvent** and in the right pane, under Event Type Details, click **Add**.

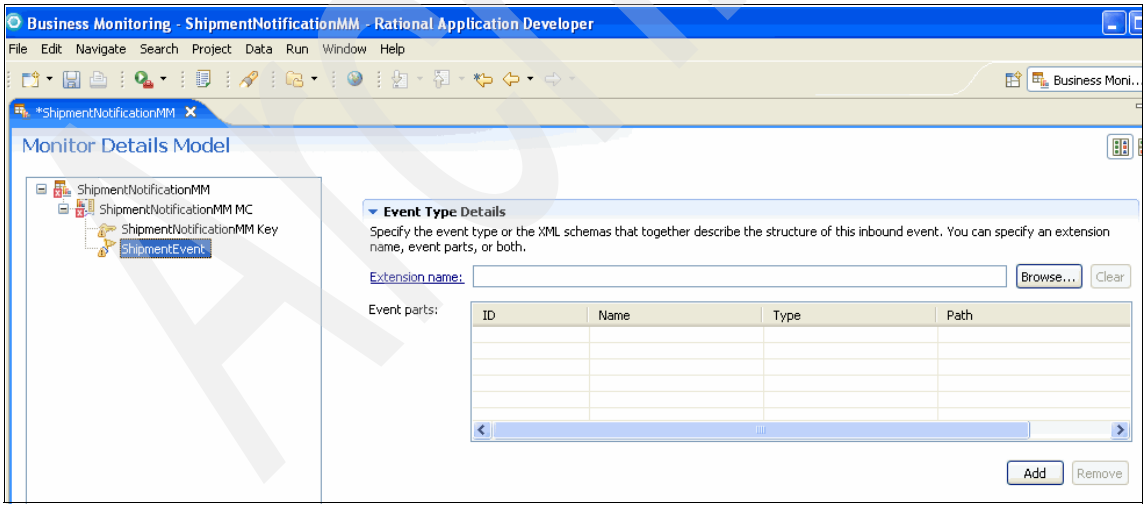


Figure 9-43 Adding event type details

Archived

4. In the Create New Event Part Type window (Figure 9-44), click **Select Type**.

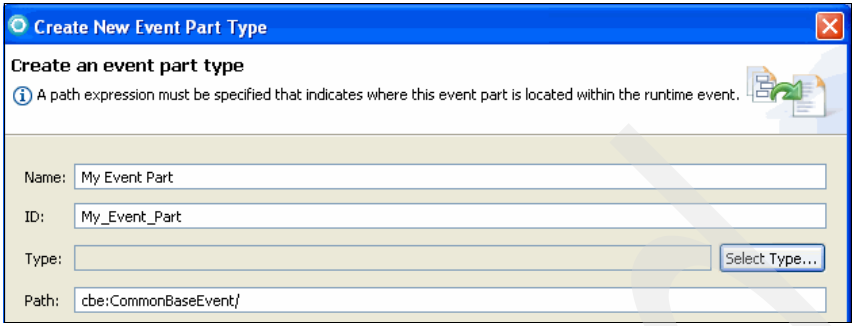


Figure 9-44 Create New Event Part Type window

5. In the Select Event Part Type window (Figure 9-45), select **Choose the data type from the XML schemas accessible from this monitor project**. Expand **FileEmitterImpl** → **src** → **ShipmentNotification.xsd** and select **tns:ShipmentNotification[tns:ShipmentNotification]**. Click **Finish**.

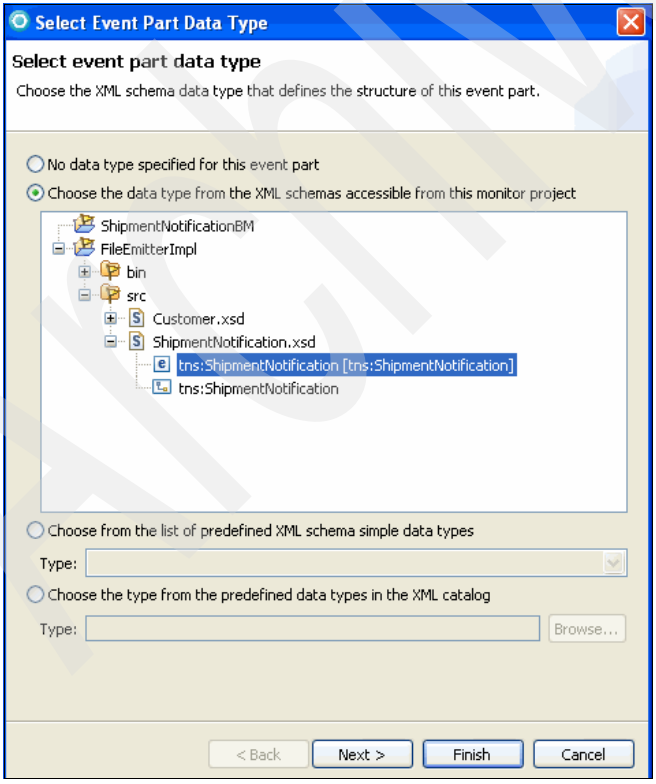


Figure 9-45 Select Event Part Type window

6. Click **Finish**.
7. Scroll down to the Correlation Expression section (Figure 9-46):
   a. In the text area, press Ctrl+Spacebar to use content assist to type `ShipmentNotificationMM_Key = ShipmentEvent/My_Event_Part /tns:OrderID`.
   b. For If no instances are found, select **Create new instance**.
   c. For If one instance is found, select **Deliver to the instance**.

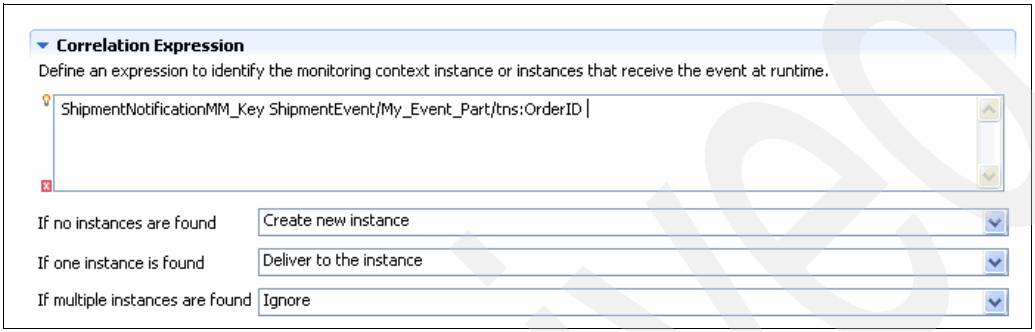


Figure 9-46 Correlation Expression

Defining a key

A *key* is a piece of information that characterizes and identifies the real-world entity that is being tracked by a monitoring context. Each monitoring context requires at least one key to make it unique. The key can be useful to correlate runtime events with the instance of the monitoring context. In this section, we define the ShipmentNotificationMM key for the ShipmentNotification MM monitoring context:

1. In the Monitor Details Model panel, in the left pane, click **ShipmentNotificationMM Key**.
2. In the right pane, under Key Value Expressions, click **Add**.
3. Click the **...** button.
4. Using content assist, type `ShipmentEvent/My_Event_Part/tns:OrderID`.

Archived

Figure 9-47 shows the completed Key Value Expressions section.

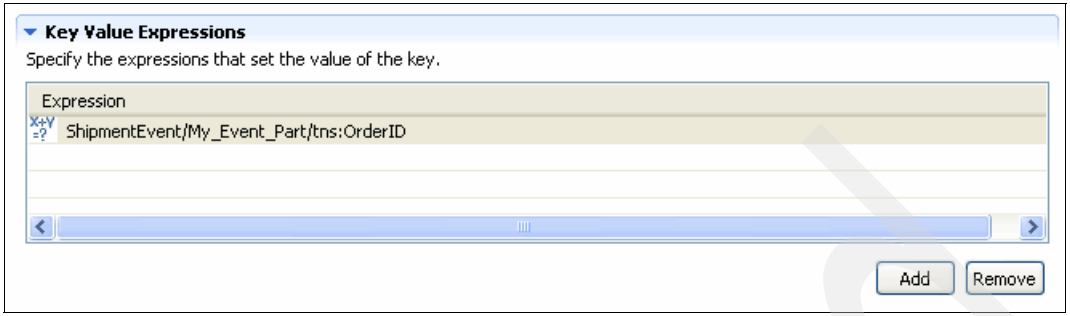


Figure 9-47 Completed Key Value Expressions

Creating a trigger

A *trigger* is a mechanism that detects an occurrence and can cause additional processing in response. In this section, we create a trigger that is fired when there is a new inbound event, with an order status that is shipped:

1. Right-click **ShipmentNotificationMM MC** and select **New** → **Trigger**.
2. For the name of the trigger, type `Order Shipped` and click **OK**.
3. Select the newly created **OrderShipped** trigger. In the right pane, under Trigger Sources, click **Add**.

Archive

4. In the Select Trigger Source window (Figure 9-48), select **ShipmentEvent** as the cause for the trigger to fire. Click **OK**.

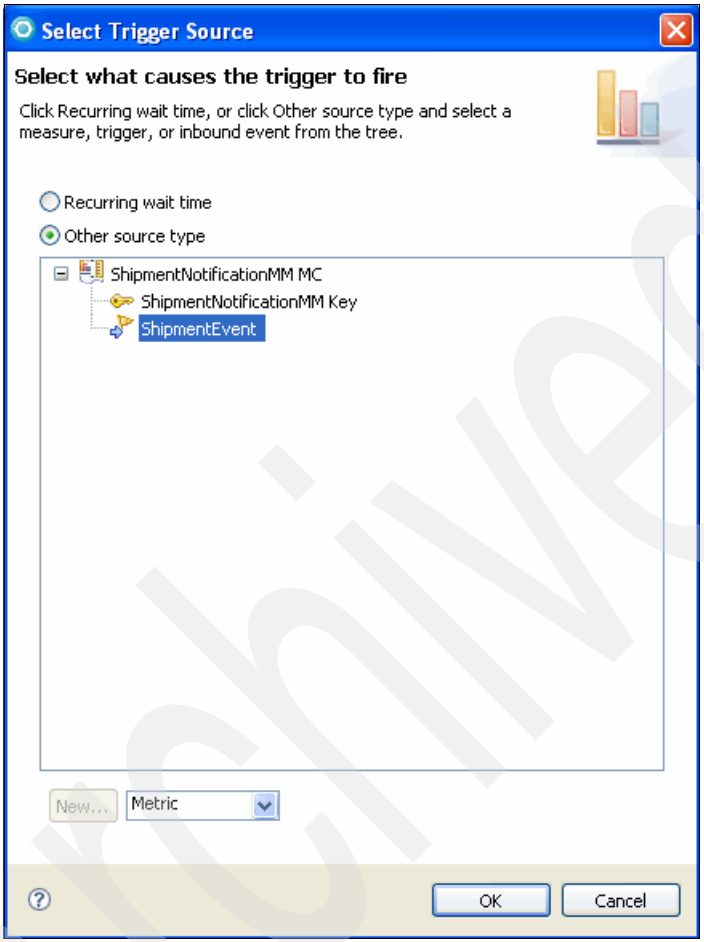


Figure 9-48 Select Trigger Source window

Archived

5. In the Trigger Condition section (Figure 9-49), use content assist to help you type `ShipmentEvent/My_Event_Part:tns:Status = 'SHIPPED'`.

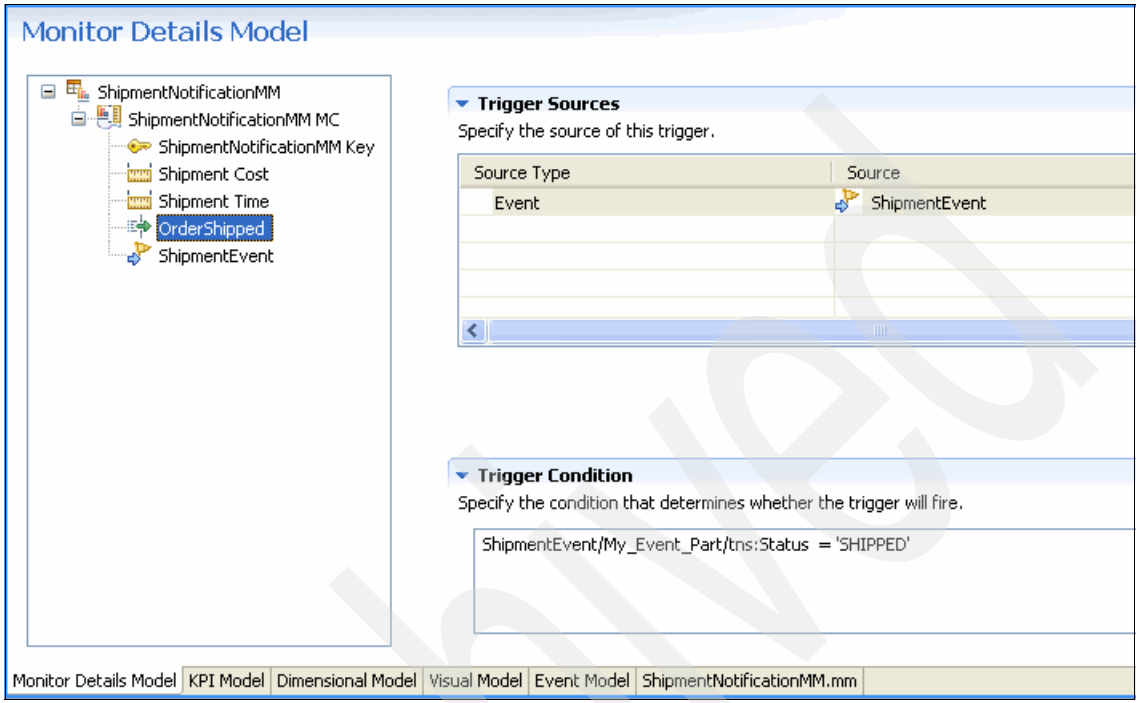


Figure 9-49 Specifying the trigger condition

6. Save your work.

> **Termination trigger:** Typically each monitoring context includes a trigger that terminates it when fired. You can create a separate trigger to terminate this monitoring context when fired.

Creating a business metric

Metrics are holders of business information. In this example, the ShipmentTime Metric holds information about the days it took to ship an order for that instance. The Shipment Cost metric holds information about the cost for shipping an order.

To create a Shipment Time metric:

1. Right-click **ShipmentNotificationMM MC** and select **New** → **Metric**.
2. For the name of the metric, type `Shipment Time`. For Type, select **Integer**. Click **OK**.

3. Select the newly created **Shipment Time** metric.
4. Under the Metric Value Expressions section (Figure 9-50) in the right pane:
   a. Click **Add**.
   b. In the Expression field, click **...** and type `ShipmentEvent/My_Event_Part/tns:ShipmentTime`.
   c. In the Trigger field, click **...**, and for the trigger type, select **OrderShipped**. Click **OK**.

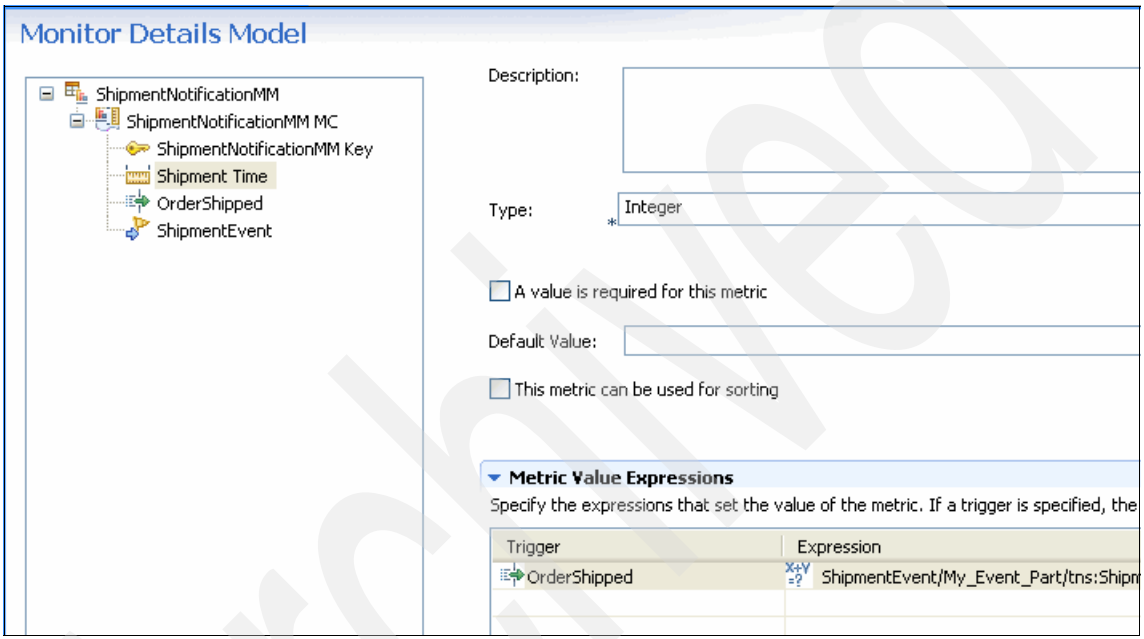


Figure 9-50 Shipment Time Metric Value Expression

To create a Shipment Cost metric:

1. Right-click **ShipmentNotificationMM MC** and select **New** → **Metric**.
2. For the name of the metric, type `Shipment Cost`. For Type, select **Integer**.
3. Select the newly created **Shipment Cost** metric.

Archived

4. Under the Metric Value Expressions section (Figure 9-51) in the right pane:
   a. Click **Add**.
   b. In the Expression field, click **...** and type `ShipmentEvent/My_Event_Part/tns:ShipmentCost`.
   c. In the Trigger field, click **...**, and for the trigger type, select **OrderShipped**. Click **OK**.

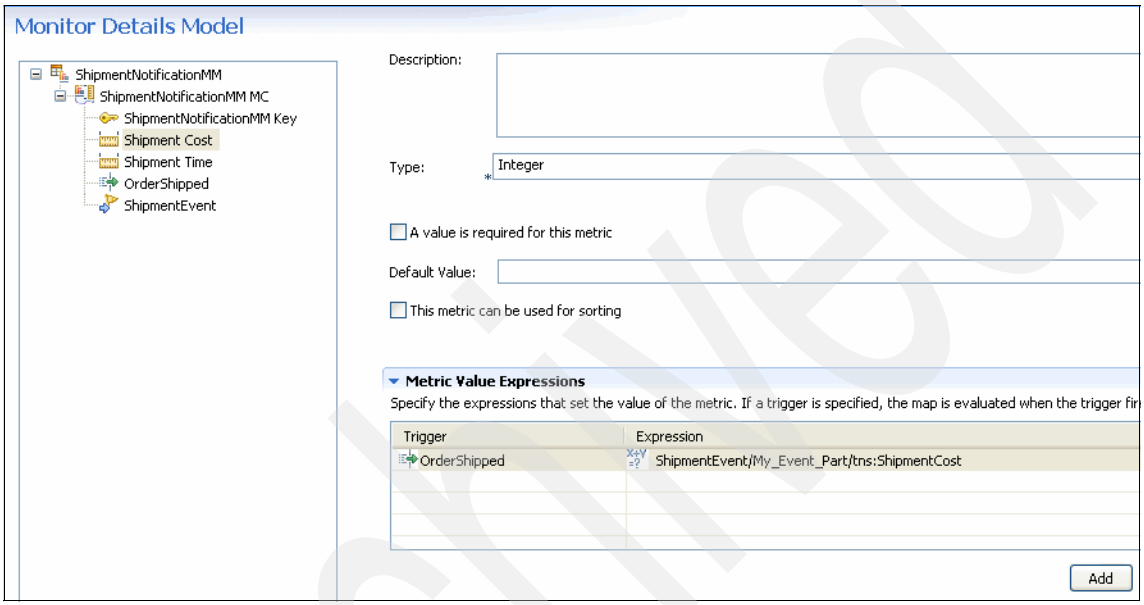


Figure 9-51 Shipment Cost Metric Value Expression

Creating a KPI

The KPIs that we create in this section are the Average Shipment Time and the Average Shipment Cost. The Average Shipment Time calculates the amount of time (in days) that it takes to ship an order on average. The Average Shipment Cost calculates the average cost incurred for shipping an order.

Archived

To create these KPIs:

1. Click the **KPI Model** tab (Figure 9-52). In the left pane, right-click **ShipmentNotificationMM and** select **New** → **KPI Context**. In the right pane, for Name, type `ShipmentKPIContext`. Click **OK**.

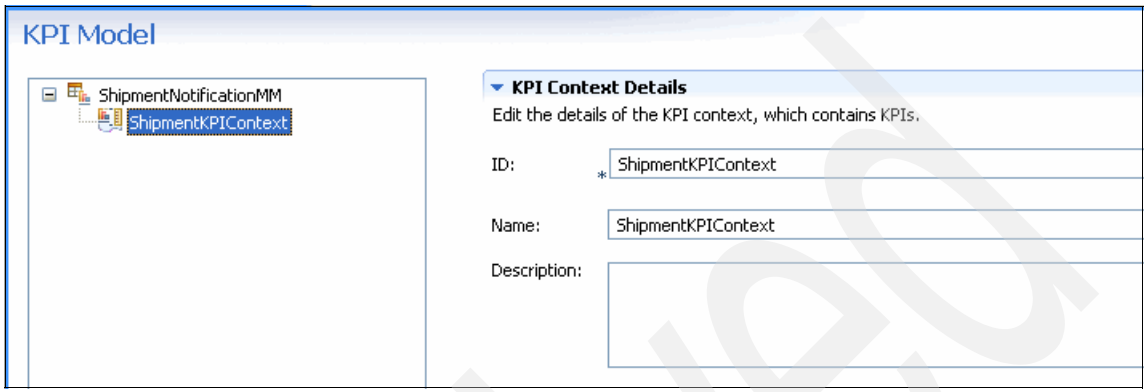


Figure 9-52 ShipmentKPIContext

2. Create the Average Shipment Time KPI:
   a. Right-click **ShipmentKPIContext and select New** → **KPI**. In the right pane, for Name, type `Average Shipment Time (Days)`. Click **OK**.
   b. Under the KPI Target and Ranges section, in the Target field, type `3`. Fill in the ranges as listed in Table 9-7.

Table 9-7 Average Shipment Time KPI target and ranges

Range name	Start value	End value
Fast	0	2
Satisfactory	2	4
Slow	4	7

Archived

Figure 9-53 shows the completed KPI Target and Ranges fields.

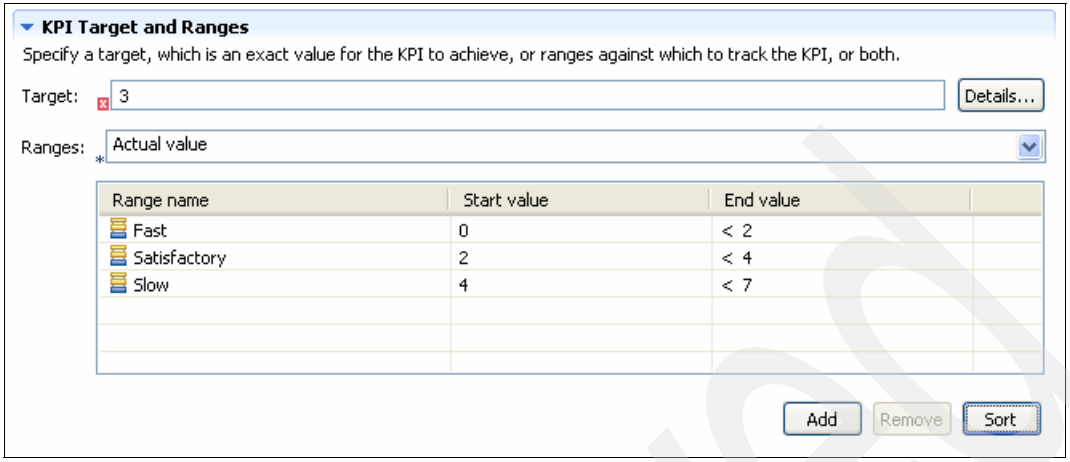


Figure 9-53 Average Shipment Time KPI Target and Ranges

c. Scroll down to the KPI Definition section.

   i. For KPI value (Figure 9-54), select **Base this KPI on a metric and an aggregation function**.

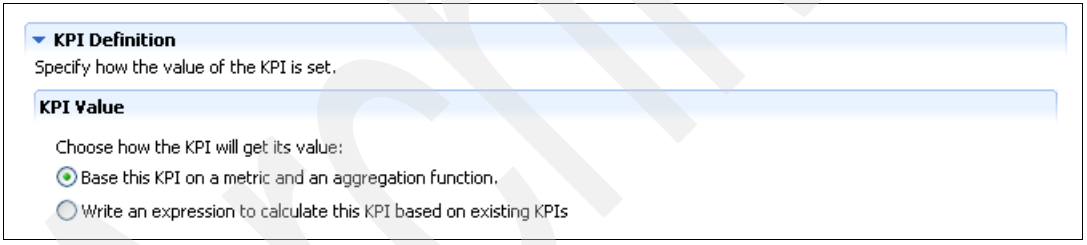


Figure 9-54 KPI Value

Archived

ii. Under KPI Details (Figure 9-55), for Monitoring context, click **Browse** and select **ShipmentNotificationMM MC**. For Metric, click **Browse** and select **Shipment Time**. For Aggregation function, select **Average**.

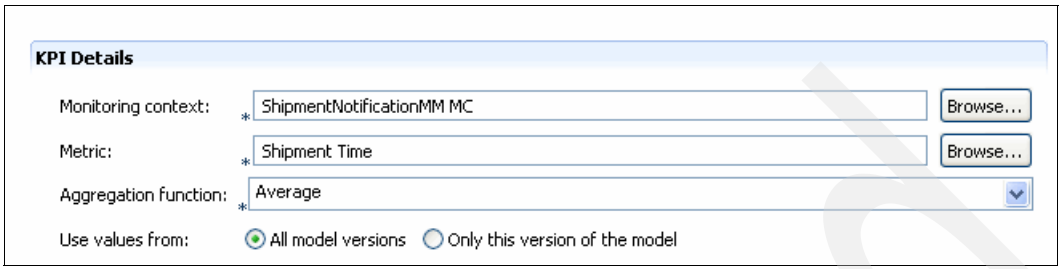


Figure 9-55 KPI Details

3. Create the Average Shipment Cost KPI:
   a. Right-click **ShipmentKPIContext** and select **New** → **KPI**. For Name, type Average `Shipment Cost (USD)`. Click **OK**.
   b. Under the KPI Target and Ranges section, in the Target field, type `30`. Fill in the range values as listed in Table 9-8.

Table 9-8 Average Shipment Cost KPI Target and Ranges

Range name	Start value	End value
Low	0	20
Moderate	20	40
High	40	60

Archived

Figure 9-56 shows the completed KPI Target and Ranges fields.

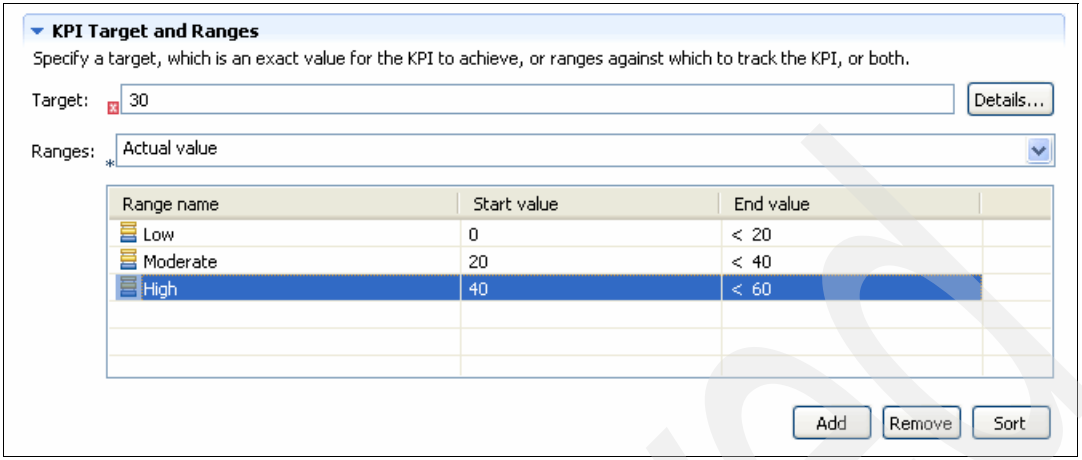


Figure 9-56 Shipment Cost KPI Target and Ranges

c. Scroll down to the KPI Definition section.

   i. For KPI value, select **Base this KPI on a metric and an aggregation function**.

   ii. For KPI Details, for Monitoring context, click **Browse** and select **ShipmentNotificationMM MC**. For Metric, click **Browse** and select **Shipment Cost** as the Metric. For Aggregation function, select **Average**.

4. Save your work.

Generating a new monitor model J2EE project

In the Project Explorer view, right-click **ShipmentNotificationMM.mm** and select **Generate Monitor J2EE Projects**. Click **Finish**.

Deploying the monitor model

To deploy the monitor model:

1. In the Server view, right-click the **WebSphere Business Monitor Server v6.1 on WebSphere Application Server** server and select **Add and Remove Projects**.
2. In the Add and Remove Projects window, select **ShipmentNotificationMMApplication** and click **Add**. Then click **Finish**.

Archive

9.7.1 Creating the dashboard

To create the dashboard:

1. In the Servers view, right-click the **WebSphere Business Monitor Server v6.1 on WebSphere Application Server** and select **WebSphere Business Monitor Dashboard**.
2. In the monitor dashboard, click the **Dashboards** tab. Click **New** to create a new dashboard.
3. For the name of the dashboard, type `ShipmentNotification` and click **OK**.
4. On the **ShipmentNotification** tab, click the **KPI** icon from the palette panel on the right and drag it to the dashboard panel.
5. Under KPIs (Figure 9-57), click the **Personalize** button.

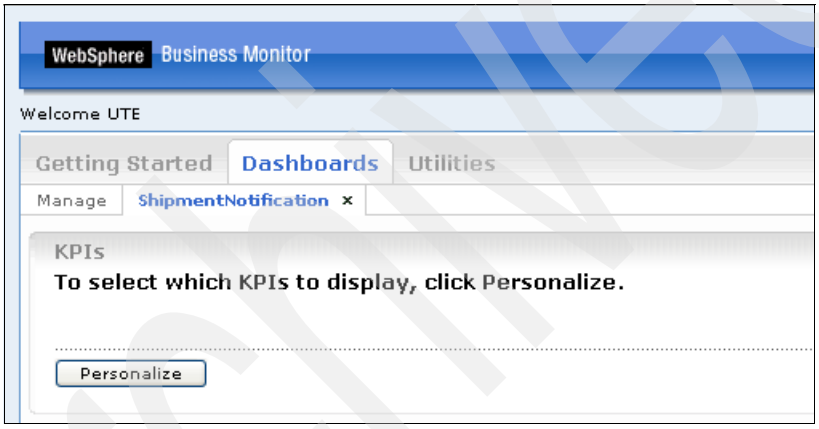


Figure 9-57 Personalizing the KPI

6. As shown in Figure 9-58, select **Average Shipment Cost** and **Average Shipment Time**, and click **OK**.

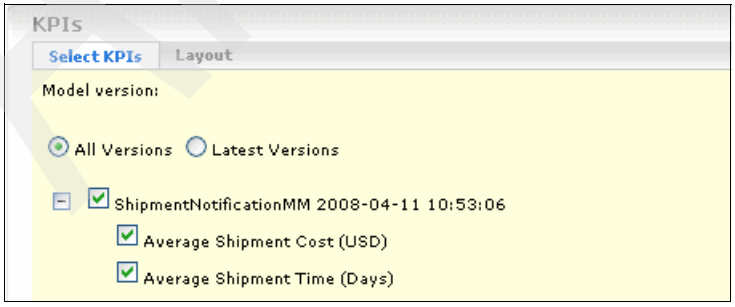


Figure 9-58 Choosing the KPIs to display

Figure 9-59 show the resultant KPI.

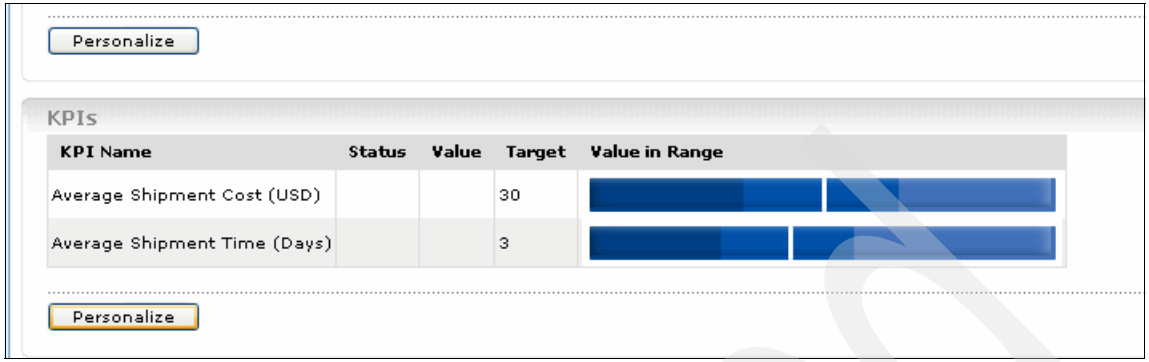


Figure 9-59 KPI portlet before running the test

9.7.2 Testing the custom event emitter sample

To test the custom event emitter:

1. Drop the sample ShipmentNotification file into the inbound directory. The scheduler picks up the file and generates the events to the Monitor server.
2. Update the dashboard as shown in Figure 9-60.

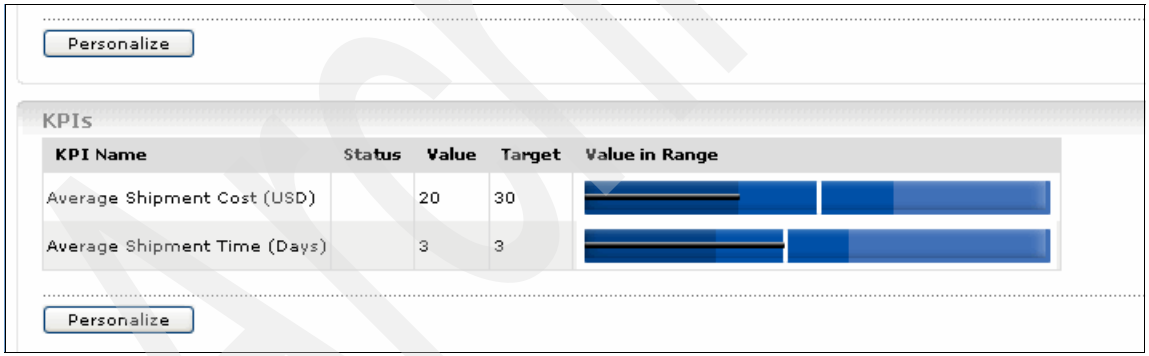


Figure 9-60 KPI portlet after running the test

Archive

3. Further customization of the portlet is possible to better reflect the KPIs captured. Figure 9-61 shows another view of the portlet.

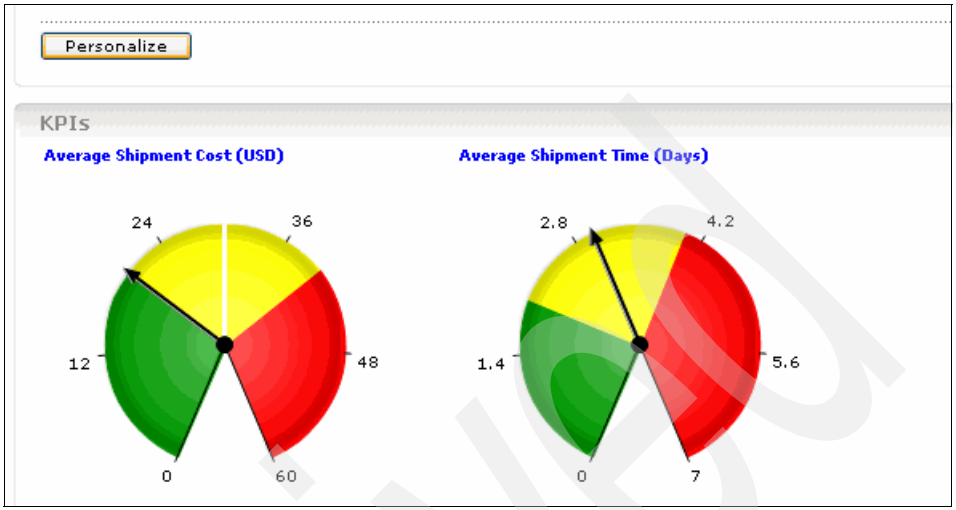


Figure 9-61 Resultant KPI

Cleaning up the dashboard

When completed, remove the KPIs from the dashboard:

1. Click the menu icon in the top right corner of the KPI portlet and select **Delete**.
2. Go to the Integration Solutions Console and uninstall FileEmitterEAR and ShipmentNotificationMMApplication.
3. Stop the server.

9.8 Design considerations

In Chapter 6, “WebSphere ESB and IBM Adapters” on page 249, we monitor events from any application by using a combination of both WebSphere Enterprise Service Bus and WebSphere Adapter. You may wonder if we can implement this scenario by using a combination of the WebSphere Flat File Adapter and WebSphere Enterprise Service Bus. The answer is *Yes*. However, this is not appropriate for several reasons, which we explore briefly in this section.

Archived

9.8.1 Custom event emitters and WebSphere Adapters

In this section, we contrast when to use a custom even emitter and when to use an Adapter within WebSphere Enterprise Service Bus for emitting events.

When to use a custom event emitter

In some environments, WebSphere Adapters, WebSphere Enterprise Service Bus, or WebSphere Process Server are not available for this purpose. For example, no suitable adapter is available. In this situation, you can use the custom event emitters as we shown in this chapter, where we deployed the emitter application on the WebSphere Application Server.

Adapters retrieve events only when there is a create, update, or delete operation on the monitored application. If events must be generated upon the occurrence of other activities in the monitored application, writing a custom event emitter offers greater flexibility in doing so.

When to use WebSphere Enterprise Service Bus with Adapters

WebSphere Adapters inherits the capabilities of the J2EE Connector Architecture (JCA) 1.5 specification as in the following examples:

- Transaction management for assured event delivery
- Connection management, which provides scalability
- Security management for end-to-end J2EE security

With WebSphere Adapter, you can immediately use these services instead of developing them separately as in the case of custom event emitters.

The use of a WebSphere Adapter also provides substantial benefits such as the reduction of development time and maintenance cost through the use of graphical tools for enterprise application discovery.

9.8.2 Creating stand-alone emitter applications or modifying existing applications

In this section, we provide a starting point for factors to consider as you decide the best suited approach for your environment.

Creating a stand-alone application

Creating a stand-alone application has the benefit of separation of concerns. The event emitter application can be modified and maintained independently from the enterprise application. The enterprise application that has gone through a period

Archived

of rigorous testing and user acceptance does not need to be modified. This approach avoids compromising the integrity of the current application.

By separating the event emitter applications from the monitored applications, a few applications can share and reuse a common event emitter, instead of having a dedicated event emitter application for individual applications.

Modifying an existing application

In some environments where only a single event source is to be monitored and there is an existing message infrastructure, it is more feasible to use the approach of modifying the existing applications to generate Common Base Events. In this case, the likelihood of reusability is low, and it might be more efficient to maintain a single application than two seperate ones.

If the application to be monitored is a new application, you can factor the generation of Common Base Events in your design as opposed to adding it later after the application has been developed.

9.8.3 Choosing Common Base Events and XSD definitions

In version 6.0.2, WebSphere Business Monitor monitored only one type of event, the Common Base Event. Although Common Base Events are not proprietary, the file format was restricted to WebSphere Business Monitor. In version 6.1, however, WebSphere Business Monitor can monitor both XSD events and Common Base Events.

XSD definitions are more robust than Common Base Event definitions. Therefore, you might prefer to use XSD event definitions if you are creating a new application. Use Common Base Events if you must monitor events that are emitted from v6.0.2 BAM or v6.0.2 Business Process Execution Language (BPEL) applications.

You might want the outbound event to be received by the action services in WebSphere Business Monitor and generate an alert in the dashboards. In this case, the event must contain an extended data element named *BusinessSituationName* with a corresponding value. A default Common Base Event named ActionServicesEvent with the required BusinessSituationName element is provided for you. Because event definition can be a mixture of Common Base Event and XSD types, you can use the provided Common Base Event and add one or more event parts in the XSD format.

9.9 Summary

In this chapter, we discussed concepts in regard to creating your own custom event emitter for monitoring abritary events. We also explained how to reuse the emitter framework and discussed the samples that are currently available for that purpose. In Chapter 10, “WebSphere MQ” on page 665, we explain how you can use an existing messaging infrastructure such as WebSphere MQ for BAM.

Archived

Archived

10

WebSphere MQ

In this chapter, we describe an example solution for a business scenario where an application is outside the managed business process but must be monitored. We explore how business events can be created and transmitted to WebSphere Business Monitor from outside the Common Event Infrastructure (CEI) by using WebSphere MQ.

The proposed solution illustrates the following concepts:

- The setup of bridging WebSphere MQ to the CEI bus infrastructure
- The building of an ad-hoc monitor model
- End-to-end monitoring of an existing application

Archived

© Copyright IBM Corp. 2008. All rights reserved.

10.1 Introduction to WebSphere MQ

WebSphere MQ is the market-leading stand-alone messaging product for IBM. It was one of the first widely used message-oriented middleware products and today remains a mainstay for many corporate messaging backbones. It is used on more than 35 hardware platforms and has API support for many programming languages. WebSphere MQ Version 7 is the most recent release.

Traditionally the WebSphere MQ model has been point-to-point messaging. In Version 5.3, publish/subscribe messaging was introduced for Java. Now, in version 7, publish/subscribe is available from C, C++, COBOL, and other application languages.

WebSphere MQ supports Java applications through the Java Messaging Service (JMS) interface to exchange messages with Java and other applications that use the WebSphere MQ API (Message Queue Interface (MQI)). The product supports transmission of text or binary messages, with individual message sizes up to 100 MB. Through message parameters, flexible topologies, message routing, and queue manager clustering, a wide range of application integration designs are possible.

WebSphere MQ supports both persistent and nonpersistent messaging. With transactional semantics and persistent messaging, messages can be recovered following a system failure. After an application sends the message, the message can only be lost if the disk on which it is stored is damaged or lost and there are no backups or mirrored copies. With nonpersistent messages, message recovery following failure is not possible, but message throughput is considerably higher.

In a typical point-to-point communication model (Figure 10-1 on page 667), the sending and receiving applications each connect to their own queue managers. WebSphere MQ is configured to route messages between queue managers by using message channels. The sending application targets a remote queue definition on its local queue manager and delivers a message. By using the configured interconnections, known as *message channels*, the queue managers exchange the message and route it to the receiving application's queue. The receiving application then retrieves the message from its queue.

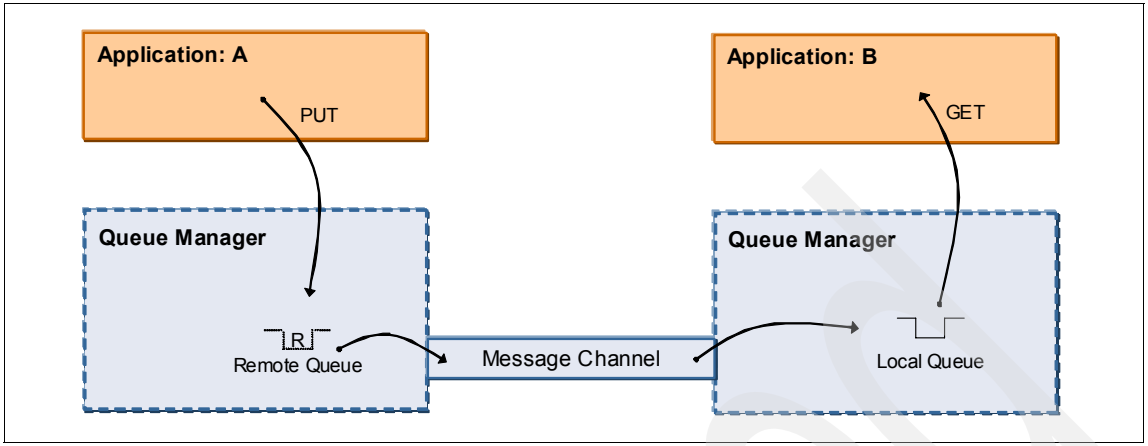


Figure 10-1 Typical WebSphere MQ topology

10.2 Overview of integrating WebSphere MQ with WebSphere Business Monitor

To understand the integration of WebSphere MQ with WebSphere Business Monitor, you must understand the technology components involved. WebSphere MQ is a messaging infrastructure product. WebSphere Business Monitor is an application that collects business events and presents them in a meaningful way to a business user. WebSphere MQ by itself does not produce business event messages, and WebSphere Business Monitor does not embody a messaging infrastructure.

When we talk about integrating the two products, we mean more than joining the two products together. At a messaging infrastructure level, we must bridge WebSphere MQ with WebSphere Application Server's CEI since that is the messaging infrastructure on which WebSphere Business Monitor relies.

At the next level, we must concentrate on how we route messages from one infrastructure to the other, targeting the appropriate destinations.

Up another level, we must discuss the entity or application in the enterprise that is creating business events and how it will do this. The events produced must be of the correct format and have meaningful definitions to ultimately be consumed by a monitor model executing within the WebSphere Business Monitor.
Figure 10-2 on page 668 illustrates the active components in this integration.

Archived

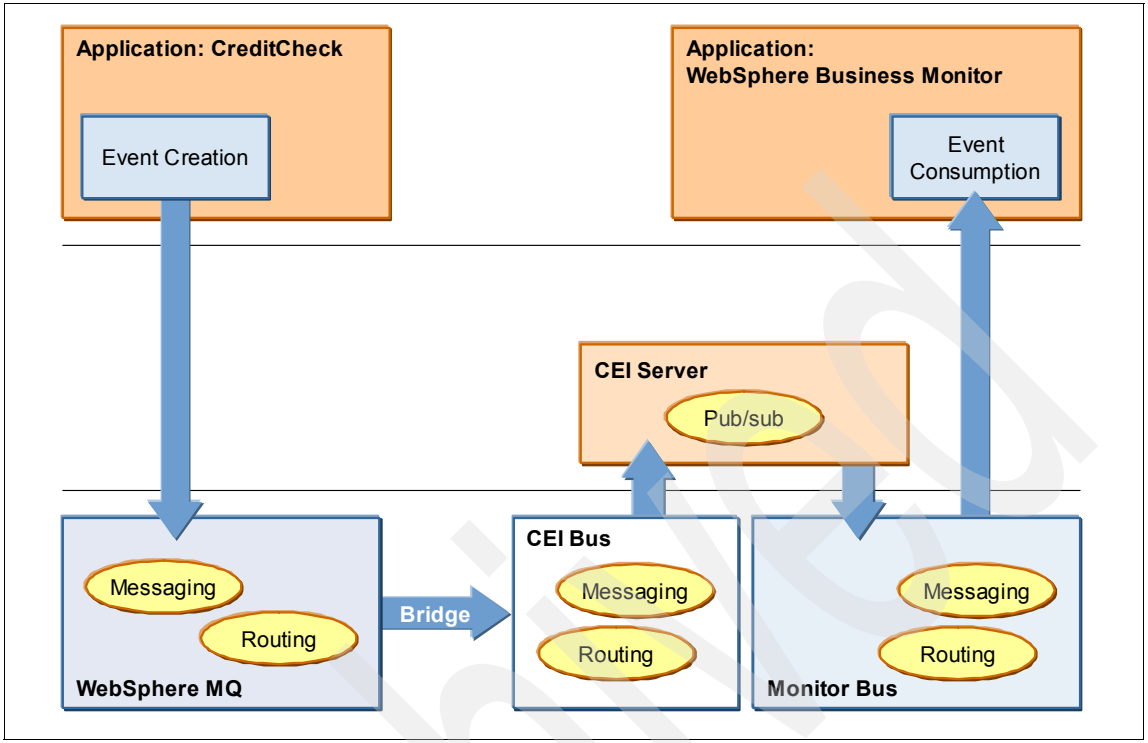


Figure 10-2 Illustration of active components

To integrate all the parts together, we have the following challenges:

- Bridge WebSphere MQ to the CEI.
- Establish routing and destination objects within both the messaging infrastructures.
- Enable a business application to format and emit business events.
- Define a monitor model that consumes those events, assembling information that can then be used to display meaningful status to the business user.

These steps illustrate a basic approach to solving the business challenge found in 10.3, “The business challenge” on page 669.

Abridged

10.3 The business challenge

In the ClipsAndTacks business scenario in Chapter 4, “Business scenario: ClipsAndTacks” on page 139, we see a process controlled and monitored in total by the technologies that make up the IBM business process management suite. However, in many customer situations, we find existing and other unmanaged applications that must be monitored.

In the following example, we explore a way to use WebSphere Business Monitor to follow the progress of an unmanaged process.

10.3.1 Outside the managed process

In this chapter, we modify the business scenario slightly for illustration purposes. We say here that the credit check process is external to the business process model described earlier. For this chapter, the credit check process is owned and maintained by the financial arm of the business by using CICS® and MQ. Figure 10-3 shows the basic design for the application.

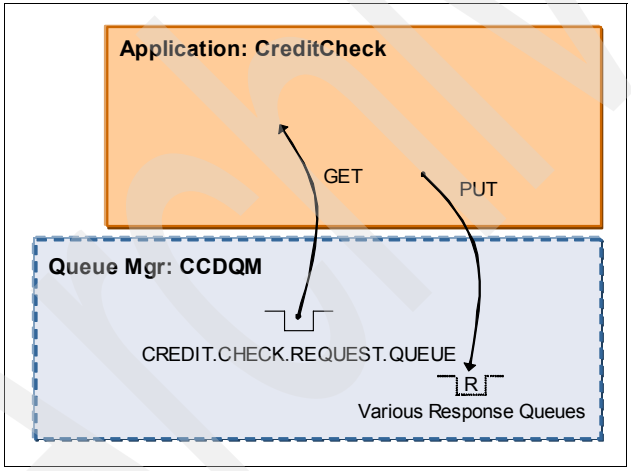


Figure 10-3 Original Credit Check application design (high level)

While the process is not managed by one of the process management solutions, it is still a critical business process. It supports requests from the managed processes and other applications throughout the business. Because it is separate from the managed processes, it will be more appropriately monitored separately.

To request the credit check service, an application formats a proprietary request message and sends it to the CREDIT.CHECK.REQUEST.QUEUE by using

WebSphere MQ. The credit check application looks up the customer data within its established list of customers and, based on the data found, approves or disapproves the sale.

To enable monitoring, the credit check application is modified to emit Common Base Events. These business events represent events points of interest in the application credit check request processing cycle. Therefore, the credit check application has a new internal component referred to as an *event emitter*. Figure 10-4 illustrates the modified application.

External event emitter: In this example, we modify the application by placing the event emitter function inside the existing application. Alternatively, an event emitter can be built to execute outside the application and emit events on behalf of the application. It creates events based on external detection of changes in the application, for example through calls to an application API or by examining the database log files of the application.

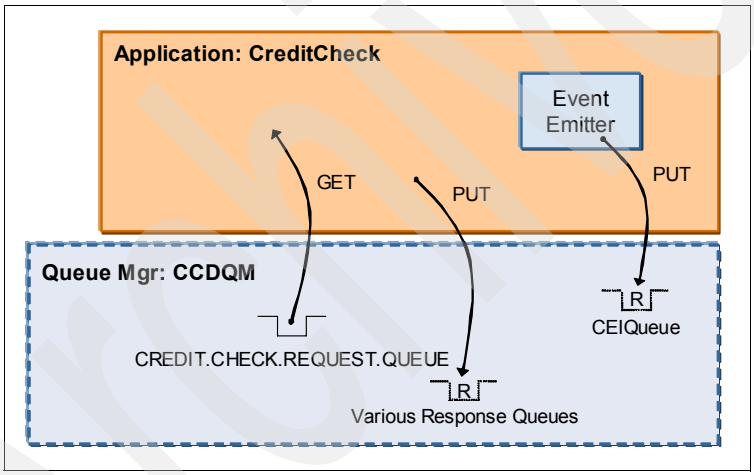


Figure 10-4 Credit Check application with event emitter added

Initially, we want to know when a credit check begins, when it is completed, and whether it provided an *Approved* status. After the monitoring mechanism is in place, the business might see the need to monitor other aspects of the process. The infrastructure allows this to happen with only changes to the events being emitted and the monitor model changed to accept them.

10.3.2 Integration topology

The financial department uses WebSphere MQ as its primary integration and application communication infrastructure. WebSphere Business Monitor uses the CEI to acquire the events it will process. The CEI is built by using the service integration bus in WebSphere Application Server. In this case, we need a way to bridge the WebSphere MQ infrastructure of the financial systems to the CEI of the business monitor.

WebSphere Business Monitor provides a script utility to set up a bridge from the CEI to WebSphere MQ. This utility uses a construct available to the service integration bus called an *MQ Link*, as illustrated in Figure 10-5.

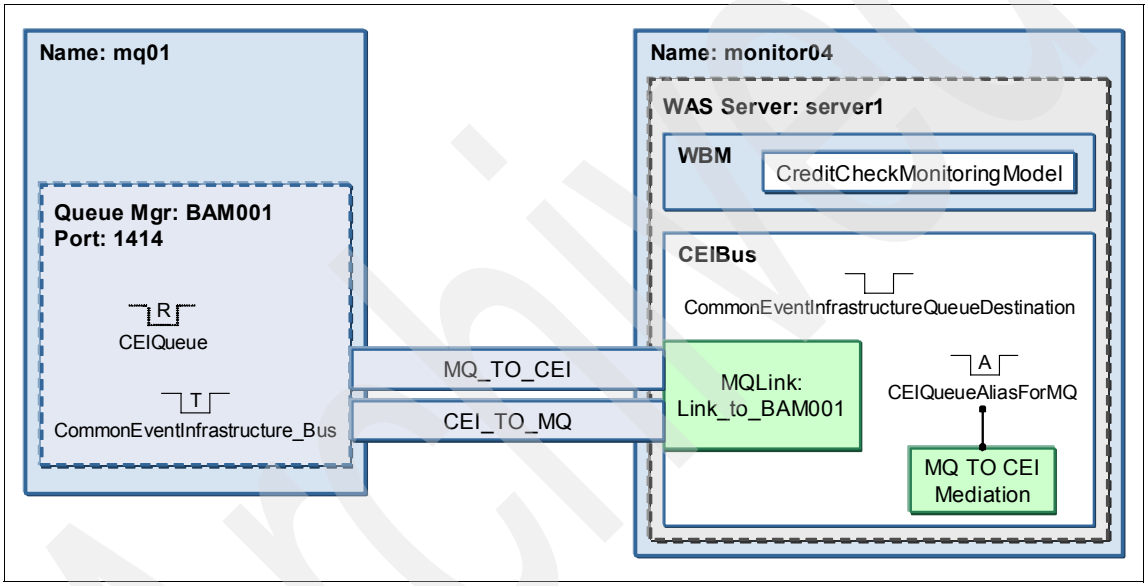


Figure 10-5 Integration of WebSphere MQ and WebSphere Business Monitor

In addition, WebSphere Business Monitor provides a mediation to adjust the inbound message slightly to allow it to be transmitted by the CEI and consumed by WebSphere Business Monitor.

The example uses the provided utilities to establish this bridge.

Archive

10.4 Building the scenario

The example assumes the following setup as a starting point:

- WebSphere Business Monitor V6.1.1 is installed on a Windows machine.
- WebSphere MQ V6.0.2 is installed on another Windows machine.
- The WebSphere Business Monitor Toolkit V6.1 is installed into an instance of WebSphere Integration Developer V6.1.

WebSphere Business Monitor is the environment where we deploy and test our monitor model. The product installation normally creates an application server profile with all the appropriate settings. We assume that this is done for this example.

WebSphere MQ is the messaging infrastructure that we want to integrate with the CEI bus on the business monitor machine. The installation does not set up a queue manager by default. Therefore, we provide the basic steps to set up a queue manager named BAM001.

WebSphere Integration Developer with WebSphere Business Monitor Toolkit is used for the development of the monitor model.

In the window captures, the WebSphere Business Monitor system is referred to as *monitor04*, and the WebSphere MQ system is referred to as *mq01*. When you attempt to reproduce the example, you may choose to use other physical system arrangements and deployment topologies with appropriate modification of the details provided.

10.4.1 Setting up WebSphere MQ

In this section, we illustrate a basic approach to setting up a queue manager by using the product's Eclipse-based management tool, WebSphere MQ Explorer. We build a queue manager and default objects for use later in the example.

To set up WebSphere MQ:

1. On the system where WebSphere MQ is installed (for example, mq01), start WebSphere MQ Explorer. From the Windows Start menu, click **Start** → **All Programs** → **IBM WebSphere MQ** → **WebSphere MQ Explorer**.
2. Create the queue manager to be used in the example. Right-click **Queue Managers** and select **New** → **Queue Manager** (Figure 10-6 on page 673).

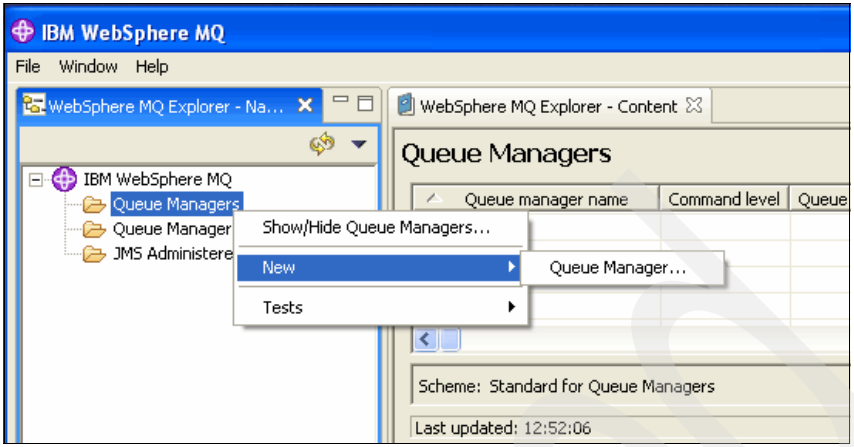


Figure 10-6 New queue manager menu navigation

3. In the Create Queue Manager window (Figure 10-7), for Queue manager name, type `BAM001` and then click **Next**.

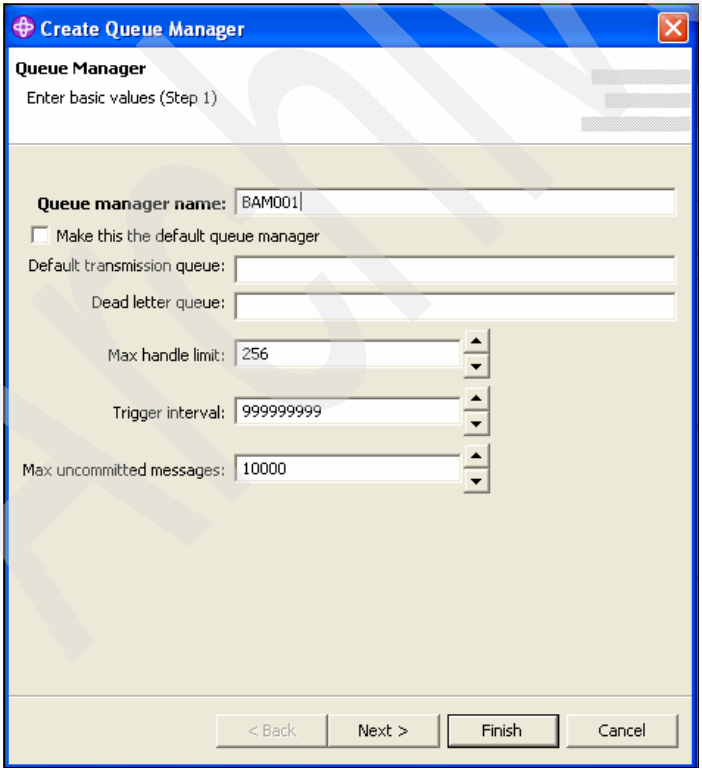


Figure 10-7 Create Queue Manager window - Queue manager name

4. In the next window (Figure 10-8), click **Next** to accept the pre-filled defaults.

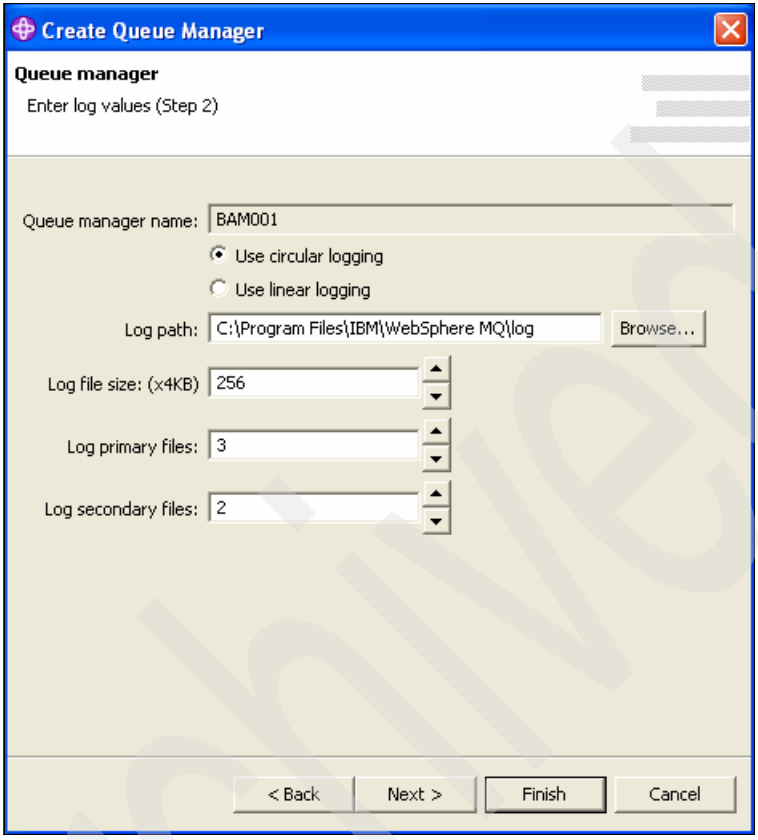


Figure 10-8 Create Queue Manager window - Log values

Archive

5. In the next window (Figure 10-9), select **Create server-connection channel** to allow for a default configuration for remote administration. Click **Next**.

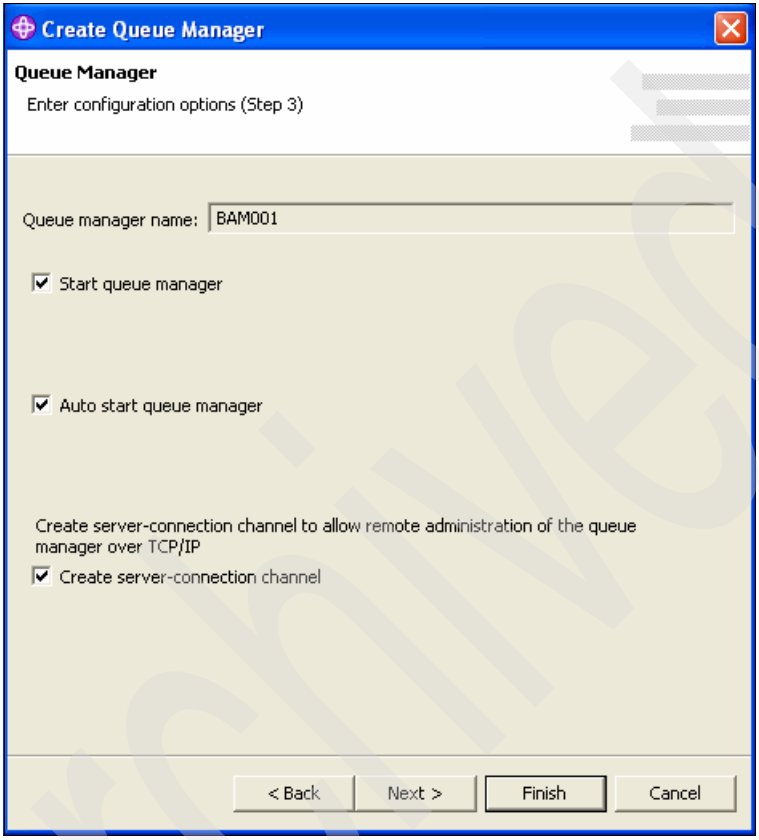


Figure 10-9 Create Queue Manager window - Create server-connection channel

Archive

6. As shown in Figure 10-10, ensure that the **Create listener configured for TCP/IP** option is selected and the port value is set to 1414. Click **Next**.

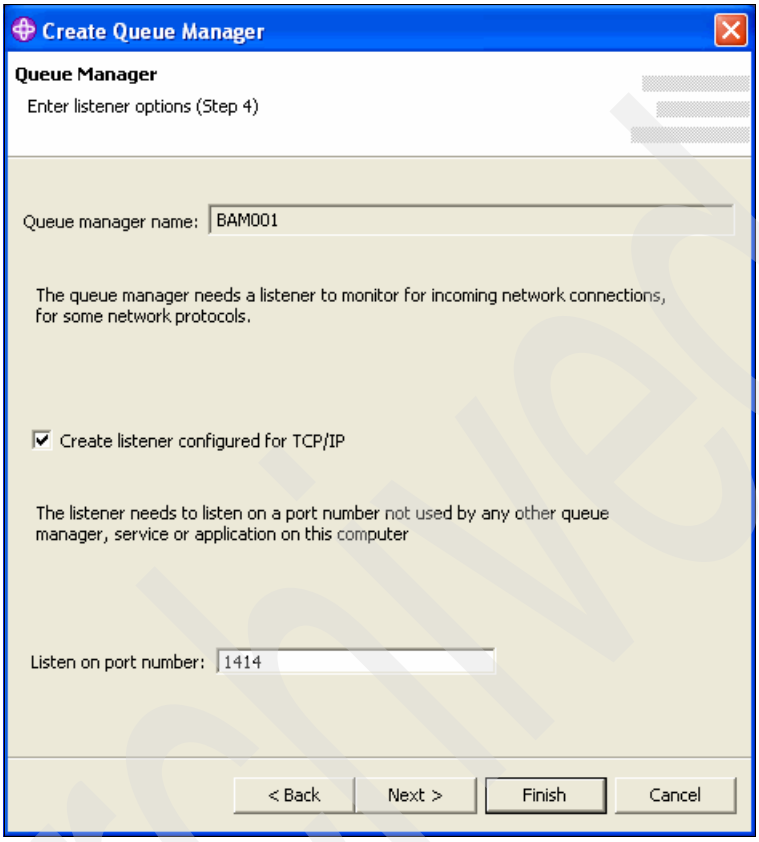


Figure 10-10 Create Queue Manager window - Listener options

Archive

7. In the next window (Figure 10-11), click **Finish** to complete the definition of the BAM001 queue manager and to start its creation.

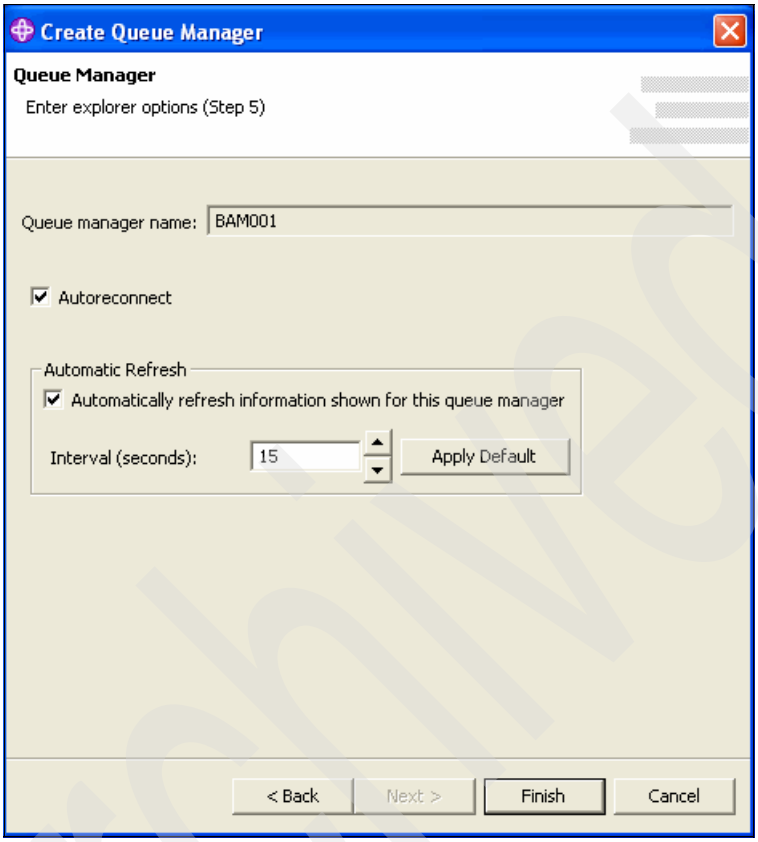


Figure 10-11 Create Queue Manager window - Completing the definition

Archived

Figure 10-12 shows how the new BAM001 queue manager is displayed in WebSphere MQ Explorer.

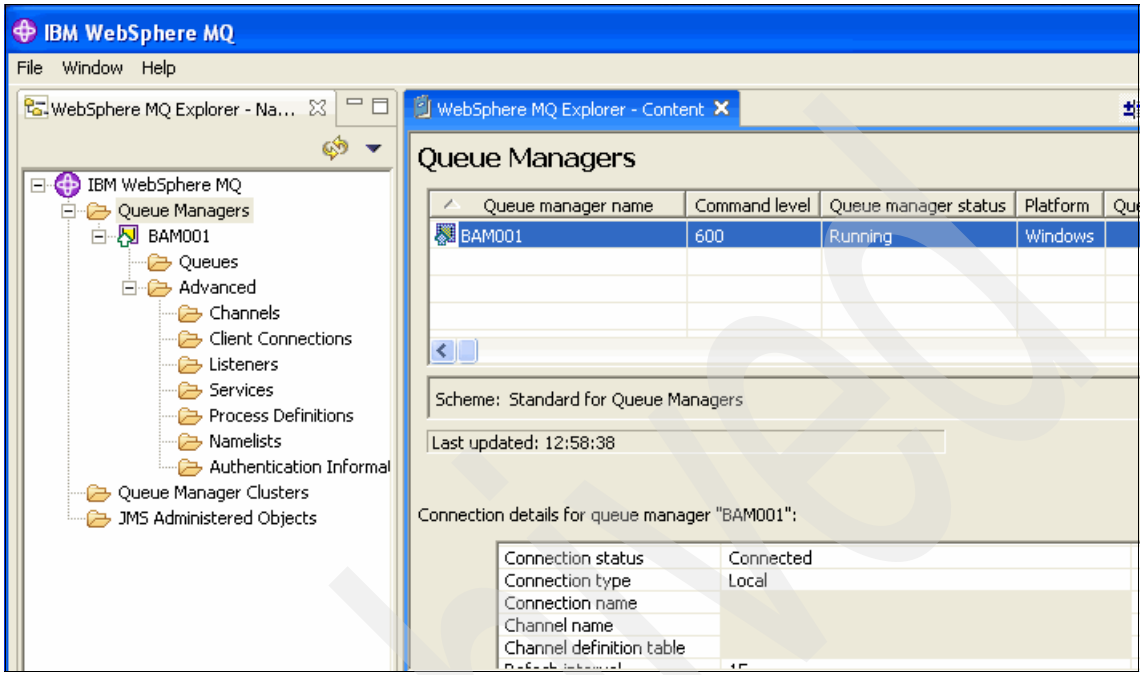


Figure 10-12 Explorer view of new queue manager

The queue manager setup activity is complete. Optionally use the WebSphere MQ Explorer to browse the configuration and objects that are setup for this queue manager. Completion of these steps provides the following definitions for use in subsequent steps:

- A started queue manager named *BAM001* running on the *mq01* system
- A started TCP/IP listener on port 1414
- A started command server for remote administration
- A remote administration channel called SYSTEM.ADMIN.SVRCONN

10.4.2 Establishing the MQ to CEI bridge

In this section, we show how to configure the bridge between WebSphere MQ and the CEI bus. We begin by installing a mediation enterprise archive (EAR) file that provides a mediation. Then we run a script to configure the interconnection pieces.

Installing the mediation EAR file

To install the provided mediation EAR file appropriately, we check on which server the CEI is configured in the monitor cell. Then using this information, we make sure the mediation is installed properly:

1. From the Integrated Solutions Console (administration console) on the system (for example, monitor04) where WebSphere Business Monitor is installed, expand **Service integration** and click **Buses** (Figure 10-13).

Figure 10-13 Service integration menu in the WebSphere Application Server administrative console

2. Click **CommonEventInfrastructure_Bus**.

Archived

3. Click **Bus Members**.

   There should be a single bus member that includes the node and server name, as shown in Figure 10-14. This node and server is where the application must be installed.

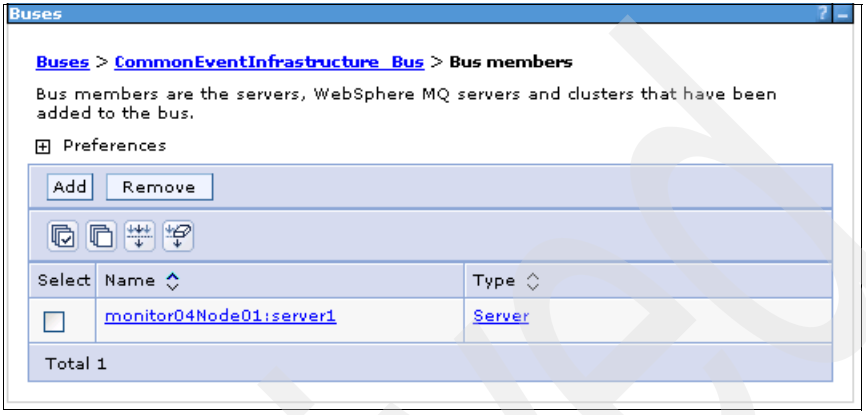


Figure 10-14 Server where the CEI is installed

4. To install the mediation application, from the administrative console in the WebSphere Business Monitor server, expand **Applications** and click **Install New Application** (Figure 10-15).

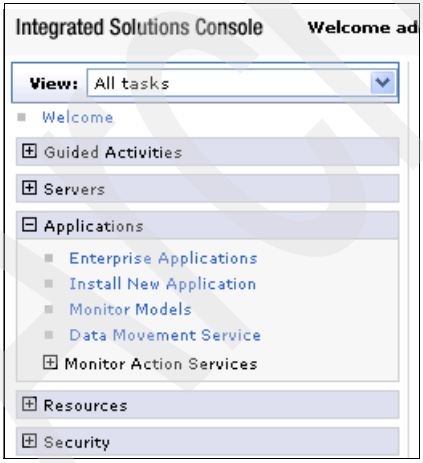


Figure 10-15 Install New Application menu

5. In the Preparing for the application installation pane (Figure 10-16):
   a. Ensure that **Local file system** is selected.
   b. For Full path, click **Browse**. Navigate to **<was_root>/scripts.wbm/CEIMQ/MQtoCEIMediation.ear**.
   c. Click **Next**.

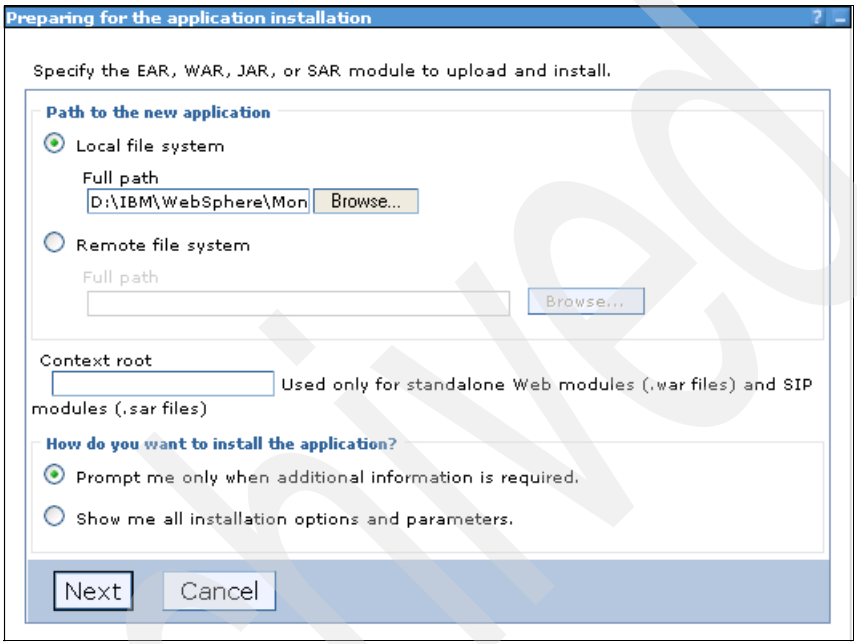


Figure 10-16 Preparing for the application installation - Path to EAR file

6. In the Select installation options panel, click **Next**.

Archived

7. In the Map modules to servers panel (Figure 10-17), make sure the server selected for the AddCEIHeaderMediation is the same as noted in step 3 on page 680. Click **Next**.

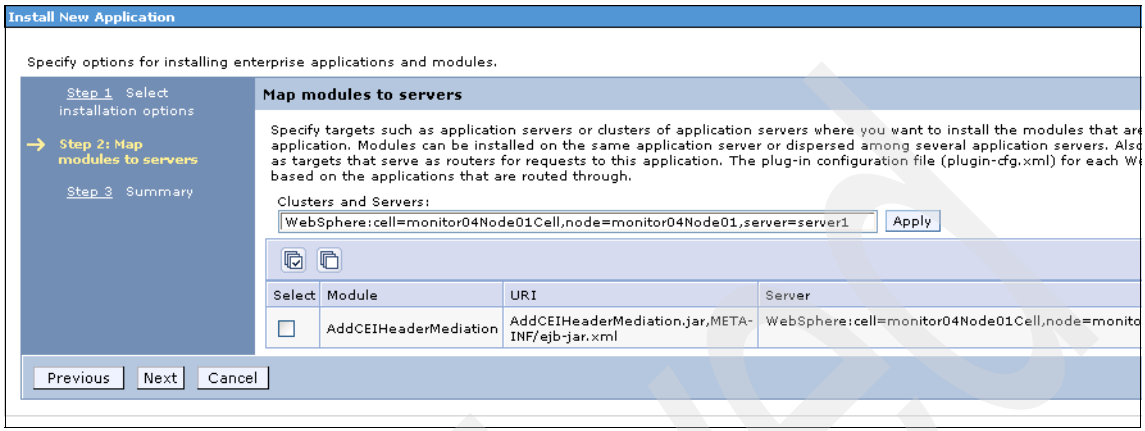


Figure 10-17 Map modules to servers panel

8. In the Summary panel, click **Finish**.
9. In the panel that shows the status of the installation (Figure 10-18), click the **Save** directly to the master configuration link.

Note: This panel shows a list of informational and warning messages concerning the installation. As long as these are all warning or informational messages, the installation is successful.

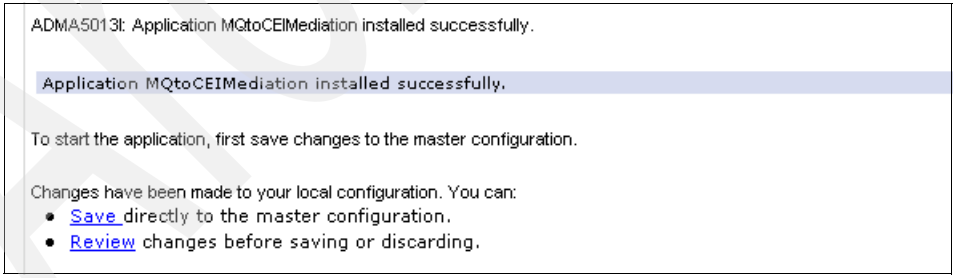


Figure 10-18 Results of the application installation

10. Make sure the mediation application is started. From the administrative console, expand **Applications** and click **Enterprise Applications**. If the newly installed application is not started, select **MQtoCEIMediation** and click **Start**.

Running the configuration script

The second part of setting up the bridge between WebSphere MQ and the CEI bus is to run the provided configuration script, which creates an object to set up the bridge:

1. Open a Command Prompt window. Navigate to the <was_root>/scripts.wbm/CEIMQ/ directory as shown in Figure 10-19. Type the name of the batch file, `configCEIForMQClients.bat`, and press Enter.

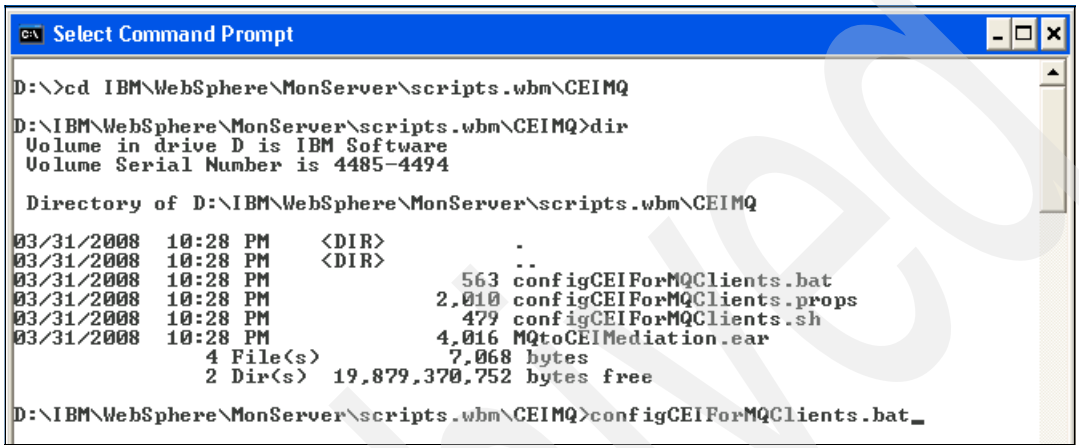


Figure 10-19 Using the command line for running the configCEIForMQClients script

The **`configCEIForMQClients.bat`** script has two modes of execution. It can be run for interactive input or it can be run by using a properties file. In this example, we run it interactively.

A choice that is given within the script is to directly connect to the MQ queue manager to configure it. Another option is to have the script generate WebSphere MQ object definitions in an MQSC script. For this example, we to have the script connect to the queue manager directly for the configuration.

Archived

2. Enter the following parameters when prompted:
   a. For Fully qualified hostname of the server hosting CEI, press Enter to accept the default.
   b. For SOAP port of the server hosting CEI, press Enter to accept the default.
   c. For Administrator or configurator userid, press Enter to indicate no security enabled.
   d. For Connect directly to WebSphere MQ, type `connect` and press Enter.
   e. For Userid for WebSphere MQ administration, press Enter to accept the default of no user ID.

      Note: An ID might be required depending on hosting environments and product setup.

   f. For Name of the administrative channel on the queue manager, press Enter to accept the default of SYSTEM.ADMIN.SVRCONN.
   g. For Name of the WebSphere MQ queue manager, type `BAM001` and press Enter.
   h. For Fully qualified hostname of the queue manager, type `mq01` (or the host name of your queue manager) and press Enter.
   i. For Port to be used for communications to the queue manager, press Enter to accept the default 1414.

   Figure 10-20 on page 685 shows the interactive script.

Archived

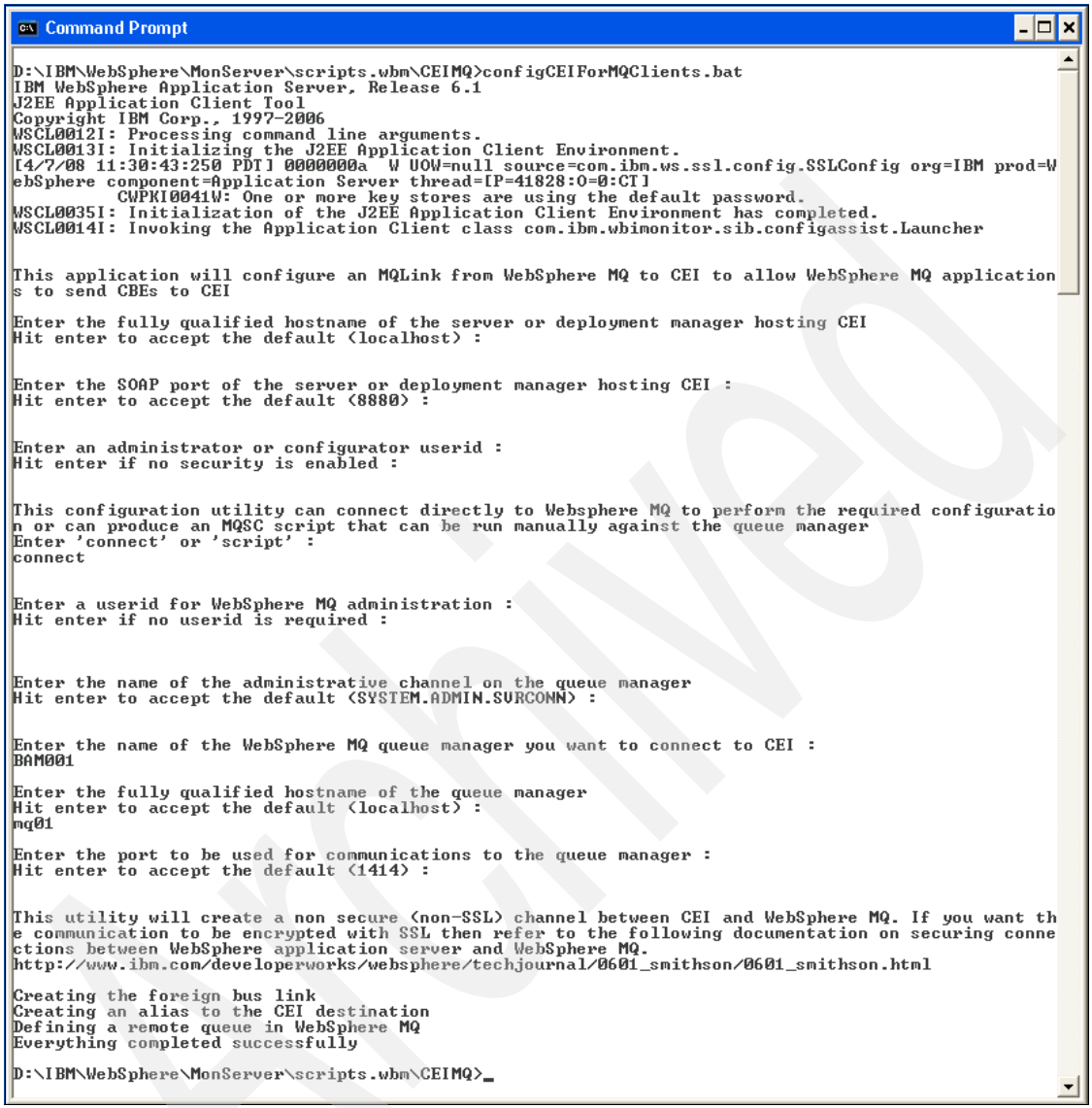


Figure 10-20 Interactive CEIMQ script

Note: Using the script interactively as described means it will open an IP connection to the target queue manager and perform the required configuration. If you experience connection problems, look for issues that prevent the connection, such as firewalls or machine name resolution.

All the artifacts are now created on both the WebSphere Business Monitor server and the WebSphere MQ queue manager. The following objects are created within the Monitor server:

- A foreign bus definition: BAM001
- An MQLink: Link_to_BAM001
- An Alias Queue Destination: CEIQueueAliasForMQ
- A mediation: MQ TO CEI Mediation

Within the queue manager, the following objects are created:

- A sender channel: MQ_TO_CEI
- A receiver channel: CEI_TO_MQ
- A remote queue definition: CEIQueue targeting the alias queue in the CEI bus
- A transmission queue: CommonEventInfrastructure_Bus

When these connections are in place, any messages that sent to the remote queue definition are forwarded to the CEI bus for publication as an event. However, if the message does not have the correct Common Base Event structure, it is discarded.

10.4.3 Creating a monitor project

In this section, we begin to create our monitor model by first creating a project within the WebSphere Business Monitor toolkit:

1. Open WebSphere Integration Developer.
2. Create a new workspace named `WMQMM01`. Make sure the Business Monitoring perspective is active.

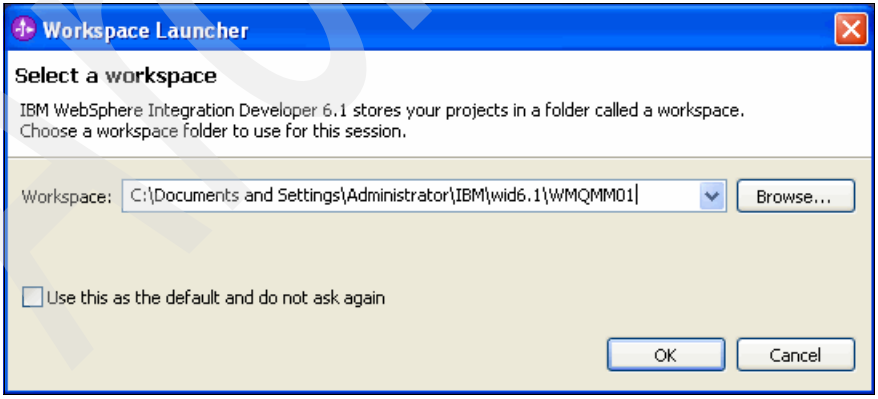


Figure 10-21 Creating a new WebSphere Integration Developer workspace

Archived

3. In the Project Explorer view (Figure 10-22), right-click and select **New** → **Business Monitoring Project**.

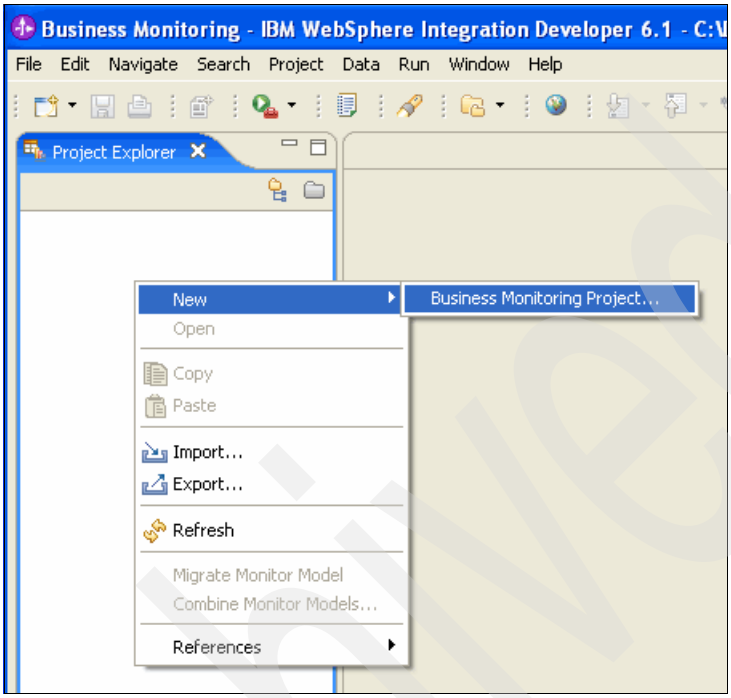


Figure 10-22 Creating a new Business Monitoring Project

Archived

4. In the New Business Monitoring Project window (Figure 10-23), for Project name, type `CreditCheckMonitoringProject`. Click **Finish**.

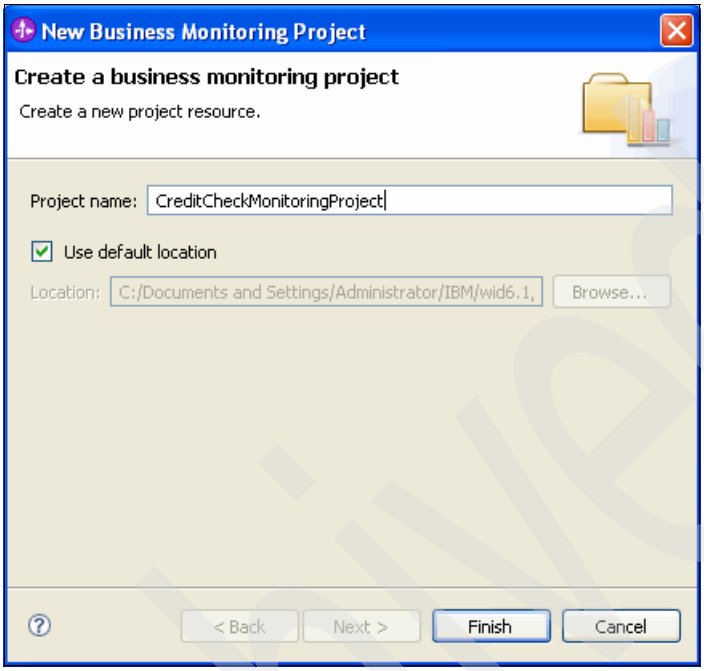


Figure 10-23 New Business Monitoring Project window

5. Examine the resulting project. Figure 10-24 shows the project contents following the creation.

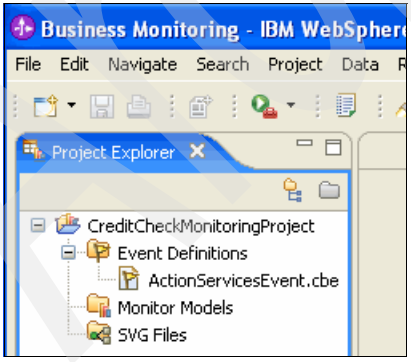


Figure 10-24 Typical monitoring project contents

The result is a new business monitoring project in which to define the business events and monitor model.

10.4.4 Defining the events

When creating a monitor model, you are required to have metadata that describes Common Base Events, XML payloads, or both that are available in the monitoring project or in a referenced project. In this way, as the various components of the model are created, the tooling can prompt for details of the event message structure.

Since our scenario involves an application that had no previous events defined, we take the approach of defining the events here. We define the events that are being emitted by the Credit Check application in a format that is meaningful to the monitor model tooling.

The Credit Check application emits two events that pertain to the monitor model, a CreditCheckRequestStarted event and a CreditCheckRequestFinished event. While the requests that initiate these events might deal with more data than represented here, for monitoring, the CreditCheckRequestStarted event must contain the following information:

- A customer number
- An order number (considered unique)
- An order amount
- A process start time stamp

Likewise, the CreditCheckRequestFinished event must contain the following information:

- An order number
- An order status
- A process end time stamp

In the example, we generate two Common Base Events customized to represent the start and end of the credit check request process. We begin defining and naming the two events in the monitoring project under Event Definitions:

1. In the Project Explorer view (Figure 10-25 on page 690), right-click the **Event Definitions** folder, select **New** → **Event Definition... (cbe)**. We use Common Base Event definitions for this example.

Archived

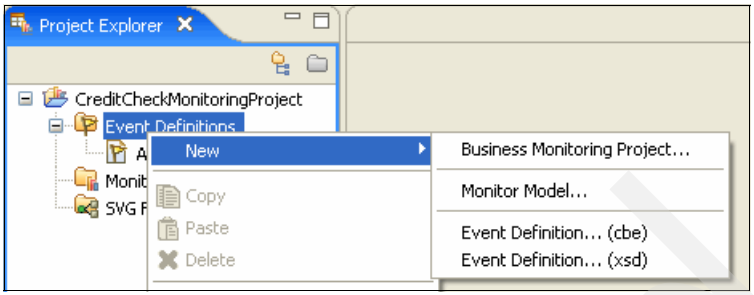


Figure 10-25 New event definition menu

2. In the New Event Definition window (Figure 10-26), for File name, type `CreditCheckRequestStart`. Click **Finish**.

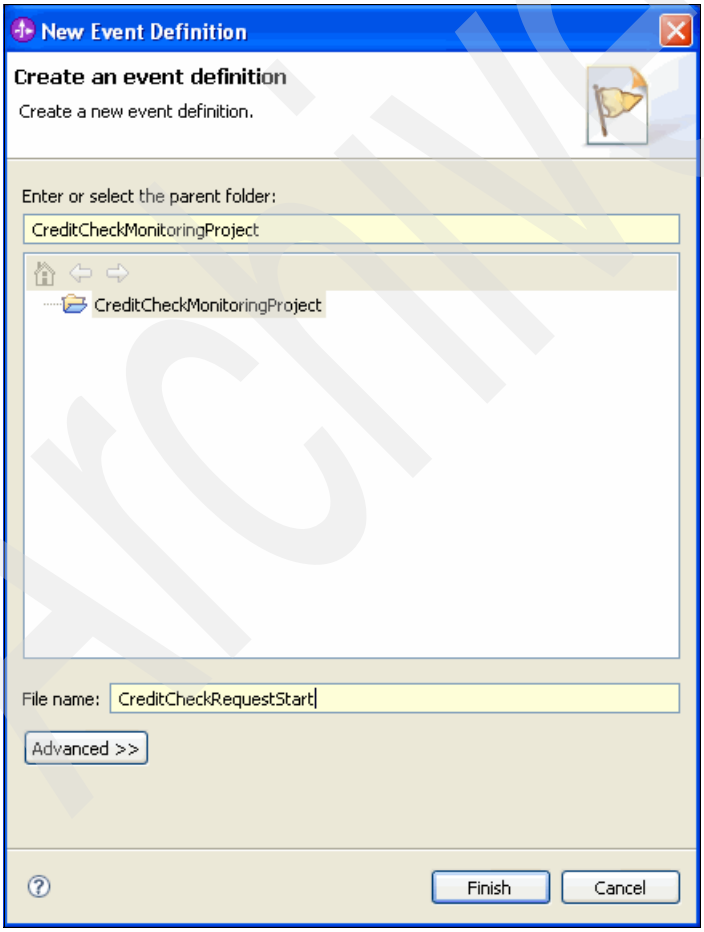


Figure 10-26 New Event Definition window

Figure 10-27 shows the results. At this point, we have an empty event definition. Notice that the Name attribute defaulted to the file name that was entered in the New Event window. This name will be used to identify an event type by the business monitor for Common Base Event processing.

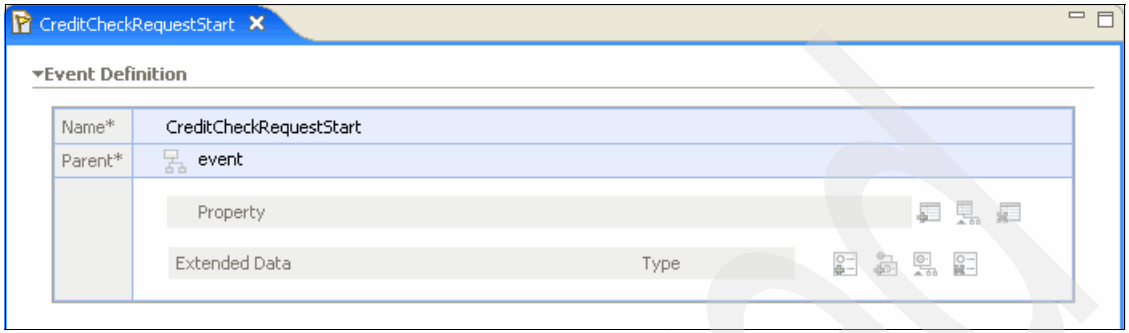


Figure 10-27 New, empty event definition

3. Create a new extended data definition to hold the event data. Click the **Add Extended Data** icon next to the Extended Data heading.

   As shown in Figure 10-28, this adds a line for an extended data holder.

4. Name the extended data definition. Click the **data1** area of the line to enter the name of the extended data element. Type `CustomerNumber` into the field.

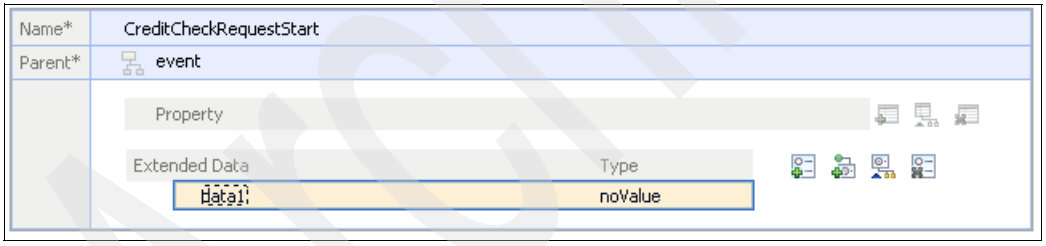


Figure 10-28 New Extended Data element

Figure 10-29 shows the result, but we still need to provide its data type.

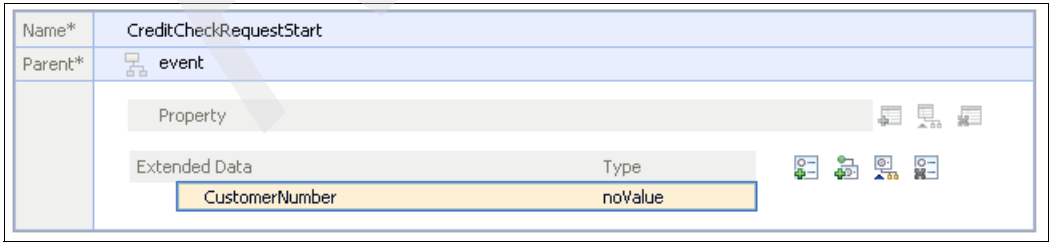


Figure 10-29 New Extended Data element with name but no data type

5. To add the data type, click the **noValue** area of the line. From the list of available data types, select **int** for CustomerNumber. Figure 10-30 shows the result.

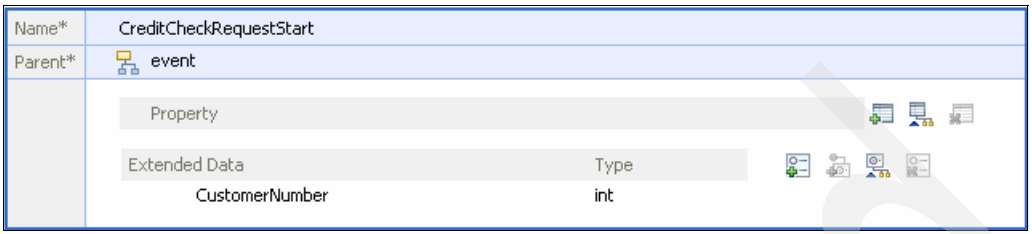


Figure 10-30 Event definition with first data element

6. In a similar fashion, add the remaining two extended data elements for the CreditCheckRequestStart event:

   a. `OrderNumber` as an `int`
   b. `OrderAmount` as a `double`

   For *start time*, we use one of the standard Common Base Event attributes, predefined in the monitor toolkit, called *creationTime*. This is reflected in the definition of the inbound events later.

   Figure 10-31 shows the results.

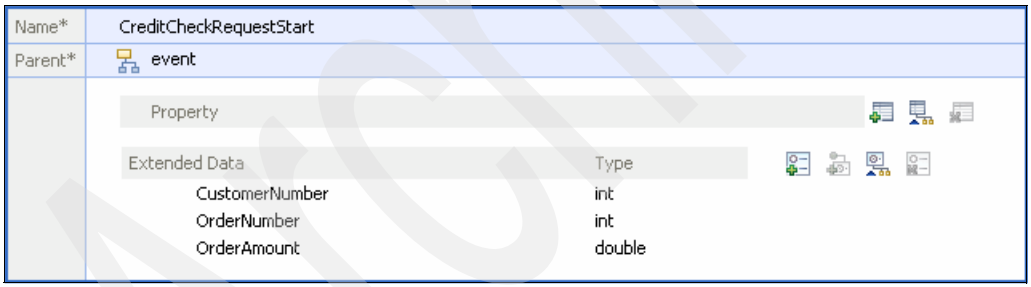


Figure 10-31 Completed CreditCheckRequestStart event

7. Press Ctrl+S or select **File** → **Save** to save your work.
8. Click **Close** to close the event definition pane.
9. Create a second event called `CreditCheckRequestFinished`. Again, use the new event definition menu as shown in Figure 10-25 on page 690. Add the extended data elements for two pieces of data:

   a. `OrderNumber` as an `int`
   b. `OrderStatus` as a `string`

Similarly, for *finish time*, we use one of the standard Common Base Event attributes, predefined in the monitor toolkit, called *creationTime*. This is reflected in the definition of the inbound events later.

Figure 10-32 shows the resulting definition.

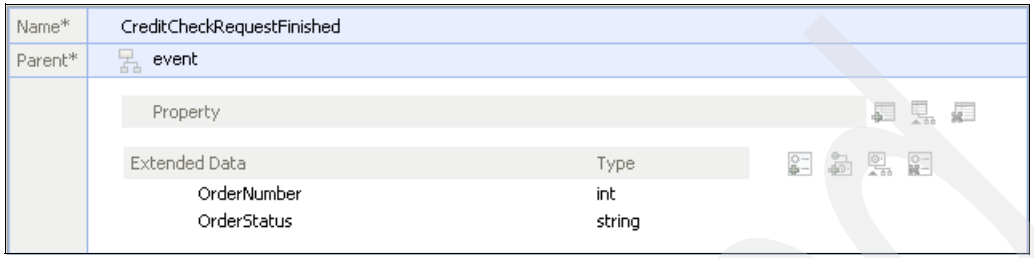


Figure 10-32 Event definition for CreditCheckRequestFinished

10. Press Ctrl+S or select **File** → **Save**.
11. Close the event definition pane.

We have now created the metadata definitions for the events that we expect to receive from the Credit Check application.

10.4.5 Defining the monitor model

Defining the monitoring model involves specifying definitions for a number of objects. In the following sections, we perform the following tasks:

- Create a monitor model
- Define the inbound events
- Define the context key
- Define the metrics
- Define any triggers

The *monitor model* holds the definitions that describe what is to be monitored. The *inbound events* indicate in which events this model is interested. The *context key* identifies the information that will be used as the unique identifier for the information in instances of the context. The *metrics* define the information to be stored within instances of the context. For both the key and the metrics, we specify how the data being stored is obtained. Finally, *triggers* indicate when to take specific actions, such as calculations or setting of metric values.

In addition, a *default KPI model* is defined to allow for an end-to-end monitoring example.

Archived

Creating the monitor model

To create the monitor model:

1. Under CreditCheckMonitoringProject (Figure 10-33), right-click **Monitor Models** and select **New** → **Monitor Model**.

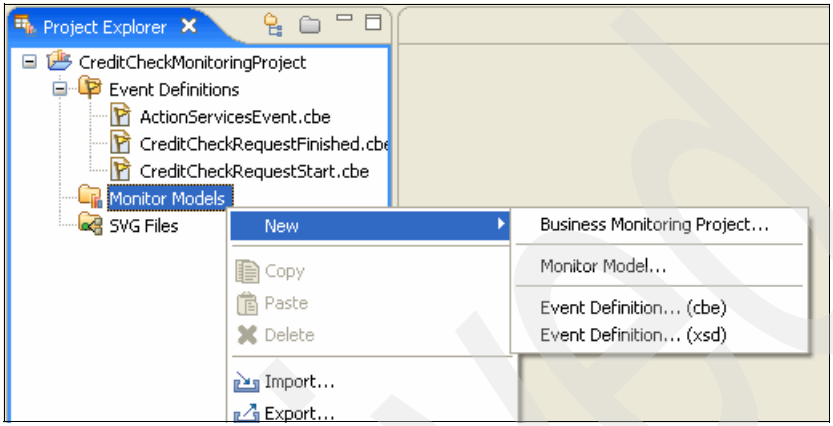


Figure 10-33 Navigating to the new monitor model

2. When prompted by the Monitor Model window, for Name, type `CreditCheckMonitorModel`. Click **Finish**. Figure 10-34 shows the new model definition.

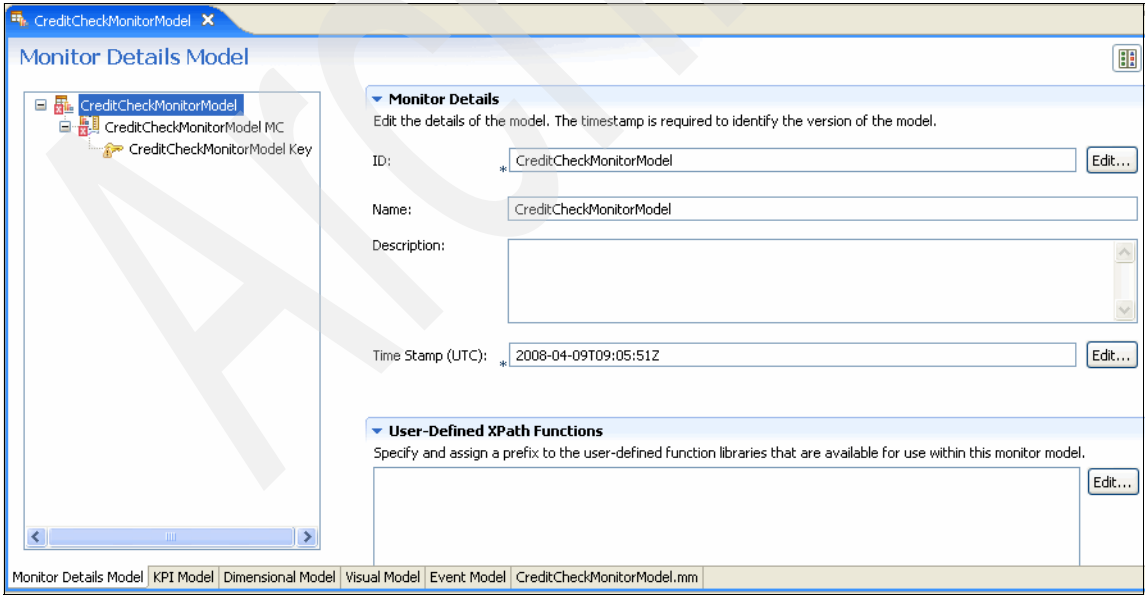


Figure 10-34 Newly created monitor model

Archived

3. To ensure the correct ordering of inbound events, set an appropriate attribute in the Event Sequence Path of the monitoring context. Click **CreditCheckMonitorModel MC**. In the Event Sequence Path section (Figure 10-35), type `*/predefinedData/creationTime`.

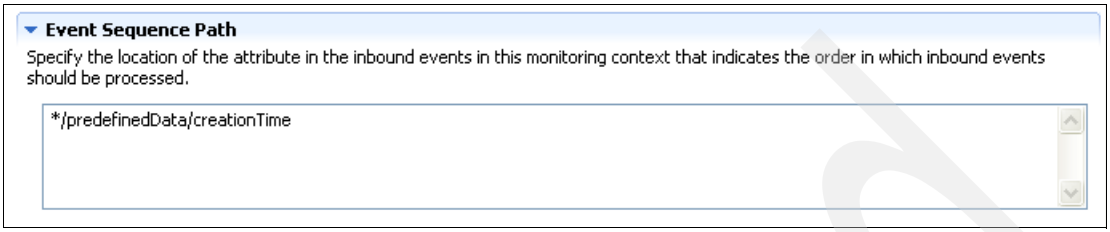


Figure 10-35 Event Sequence Path section

> **Event sequence path:** Specifying an event sequence path can be important to event processing when using asychronous communication. Many of the infrastructure components are multithreaded. It is possible for events to be processed out of sequence simply due to message queuing and multithreading.

4. Pressing Ctrl+S or select **File** → **Save** to save the monitor model.

In the following steps, we fill in the parts needed for the example.

Defining the inbound events

To indicate the events that are of interest to the monitoring context, we define inbound events in the Monitor Model editor based on the raw events defined earlier. These are subscription points or entry points for inbound events in a monitoring context. WebSphere Business Monitor subscribes to the events that you specify and delivers them to all event entry points with a matching event subscription. Each inbound event must refer to a particular event definition that defines its structure.

For the example, we define two inbound events, CreditCheckRequestStart and CreditCheckRequestFinish.

Defining CreditCheckRequestStart

To define the CreditCheckRequestStart inbound event:

1. In the Monitor Details Model, right-click **CreditCheckMonitorModel MC** and select **New** → **Inbound Event**.
2. In the Create New Inbound Event window (Figure 10-36), for Name, type `CreditCheckRequestStart`. Click **OK**.

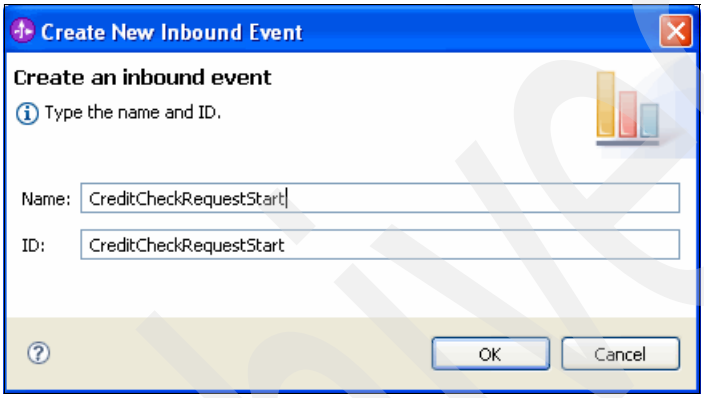


Figure 10-36 Create New Inbound Event window

3. Scroll to the Event Type Details section (Figure 10-37). For Extension name, click **Browse** to select an event.

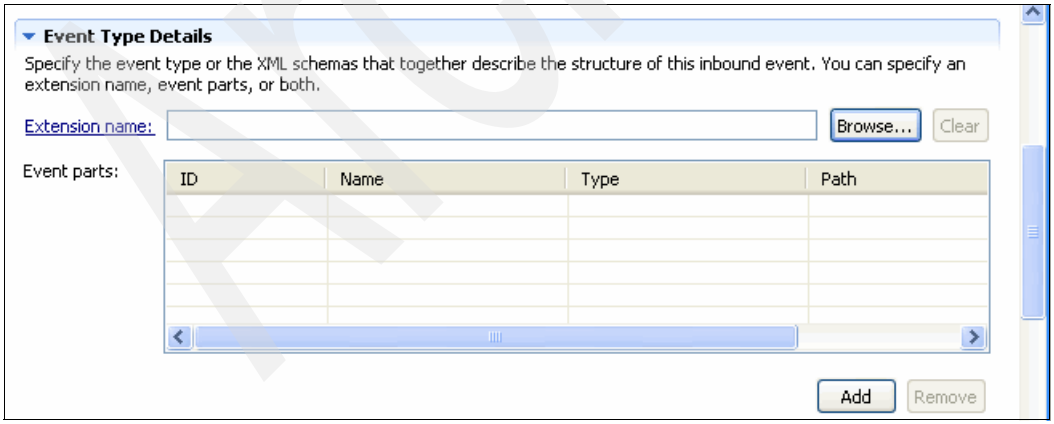


Figure 10-37 Event Type Details section

4. In the Select event definition window (Figure 10-38), select the **CreditCheckRequestStart** event type and click **OK**.

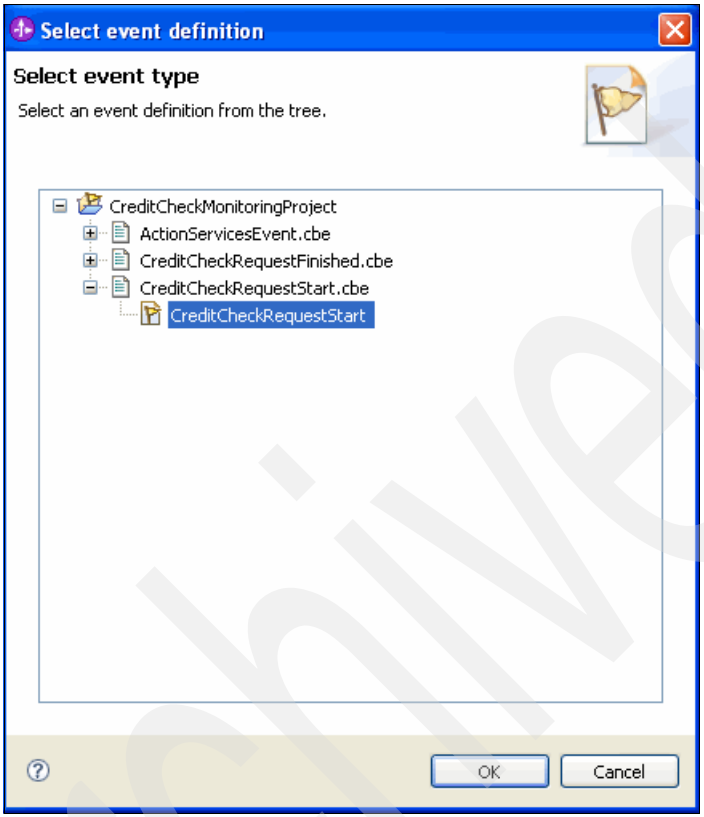


Figure 10-38 Select event definition window

Archived

5. Scroll to the Correlation Expression section of the inbound event definition (Figure 10-39):
   a. Click inside the Correlation Expression definition text box and press Ctrl+Spacebar to be prompted by an expression builder.
   b. Use the expression builder to correlate the inbound order number with the monitoring context key. The final expression is `CreditCheckRequestStart/extendedData/OrderNumber = CreditCheckMonitorModel_Key`.
   c. Set the action to take on correlation:
      i. For no instances found, select **Create new instance**.
      ii. For one instance found, select **Treat as error**.
      iii. For multiple instances found, select **Treat as error**.

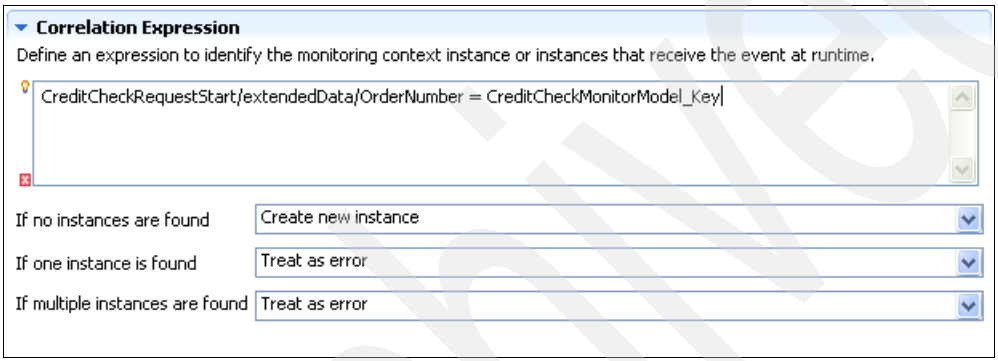


Figure 10-39 Correlation expression and conditions

The definition for the CreditCheckRequestStart inbound event is now completed.

Defining CreditCheckRequestFinished

Now define the CreditCheckRequestFinished inbound event similar to the manner in which we did for the CreditCheckRequestStart inbound event:

1. Right-click **CreditCheckMonitorModel MC** and select **New** → **Inbound Event**.
2. In the Create New Inbound Event window, for Name, type `CreditCheckRequestFinished` and click **OK**.
3. Under the Event Type Details section, set the Extension Name to `CreditCheckRequestFinished`.

Archived

4. Scroll to the Correlation Expression section of the inbound event definition (Figure 10-40):
   a. Click inside the Correlation Expression definition text box and press Ctrl+Spacebar to be prompted by an expression builder.
   b. Use the expression builder to correlate the inbound order number with the monitoring context key. The final expression is `CreditCheckRequestFinished/extendedData/OrderNumber = CreditCheckMonitorModel_Key`.
   c. Set the action to take on correlation:
      i. For no instances found, select **Treat as error**.
      ii. For one instance found, select **Deliver to the instance**.
      iii. For multiple instances found, select **Treat as error**.

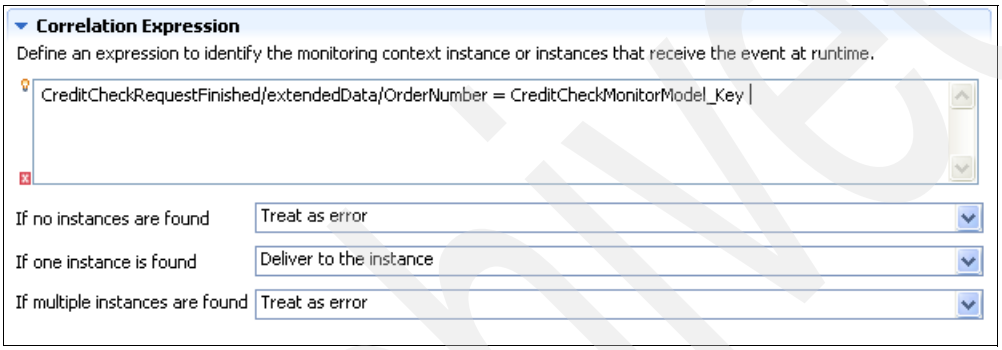


Figure 10-40 Correlation expression definition

You have now defined the two inbound events.

Error message: An error indication is present for both inbound events. This is caused by a type mismatch with the default key definition. This is corrected in the next task.

Defining the key

Each instance must be uniquely identified. When a monitoring context is created, it is given a default key definition. However, another way to think of it is to realize that a *key* is a special metric with additional attributes.

For the example, the key data value from the inbound business event is the Order Number. We change the name of the default key (to reflect its content) and store the Order Number in the key as a metric:

1. In the Monitor Details Model panel, click **CreditCheckMonitorModel MC** to view its properties.
2. In the Key Details section (Figure 10-41), modify the definition. For Name, type `OrderNumber Key`, and for Type, select **Integer**.

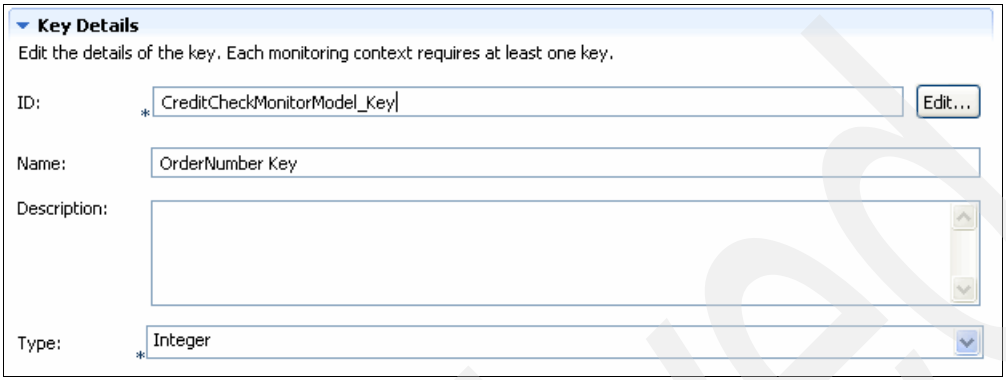


Figure 10-41 Key details

3. Scroll down to the Key Value Expressions section (Figure 10-42):
   a. Click **Add**.

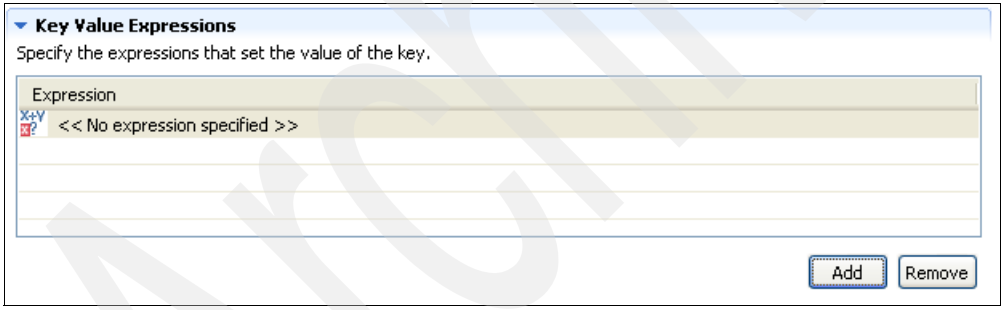


Figure 10-42 Key Value Expressions section with new expression added

Archived

b. Click in the Expression column and press Ctrl+Spacebar to be prompted with an expression builder (Figure 10-43). Expand **CreditCheckMonitorModel** → **CreditCheckMonitorModel MC** → **CreditCheckRequestStart** → **Extended Data** and select **OrderNumber**.

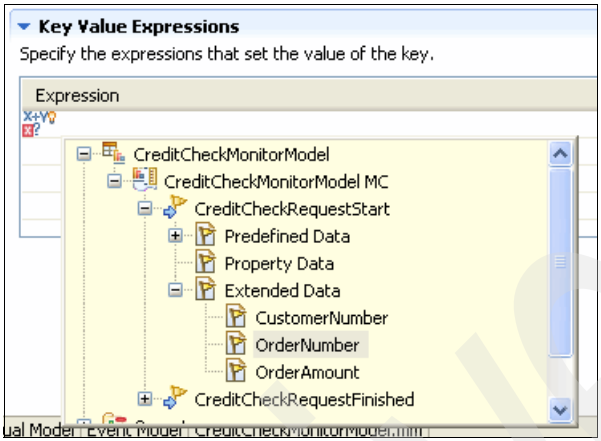


Figure 10-43 Expression builder for the key value source

Figure 10-44 shows the results in the Key Value Expressions section.

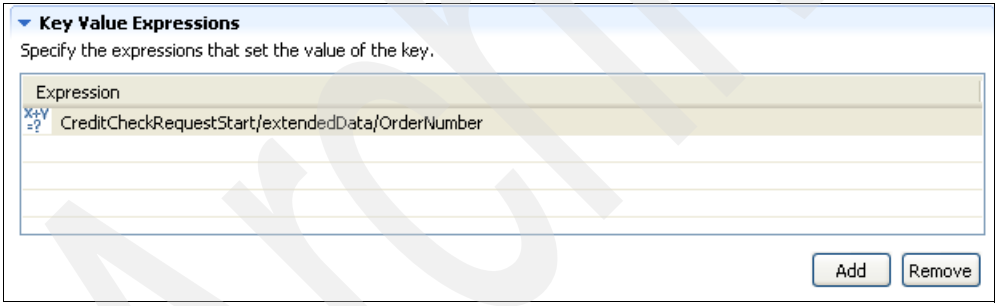


Figure 10-44 Key value expression

4. Save the monitor model.

Archived

Figure 10-45 shows the resultant monitor model content without any error messages.

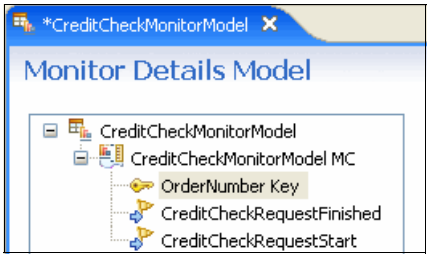


Figure 10-45 Monitor Details Model contents

Defining the trigger

Triggers define conditions upon which particular actions in the monitoring model take place. For this example, we need one trigger to fire when a CreditCheckRequestFinish event arrives to calculate the duration separately from the arrival of the data in the event:

1. Right-click **CreditCheckMonitorModel MC** and select **New** → **Trigger**.
2. In the Create New Trigger window (Figure 10-46), for Name, type `CreditCheckRequestFinished Trigger`. Click **OK**.

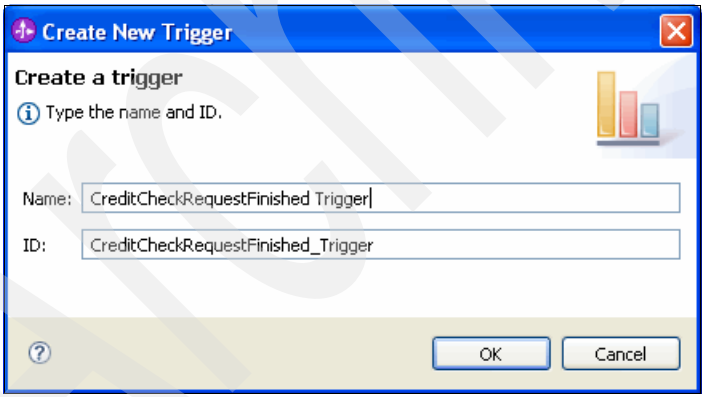


Figure 10-46 Create New Trigger window

Archived

3. In the Trigger Details section (Figure 10-47), select the **Terminate monitoring context** option.

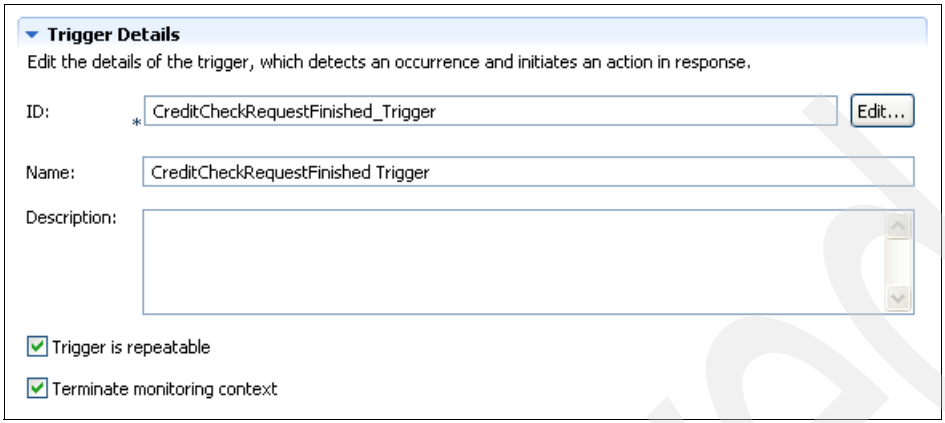


Figure 10-47 Trigger Details section

4. Scroll down to the Trigger Sources section:
   a. Click **Add**.
   b. In the Select Trigger Source window, click the inbound event **CreditCheckRequestFinished**. Click **OK**.

   This results in the trigger source definition shown in Figure 10-48. This definition indicates that the trigger fires when the inbound event occurs.

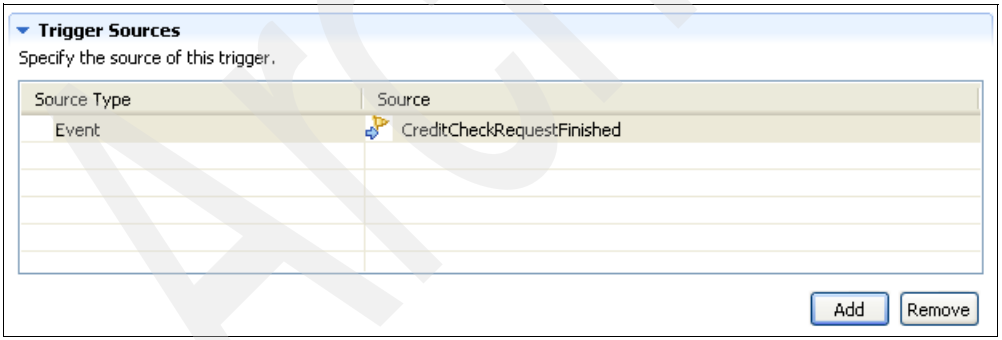


Figure 10-48 Trigger Sources section and completed source definition

The trigger source definition provides the one trigger needed in this example. It is used when defining the Duration metric in the following section.

Archived

Defining the metrics

Metrics are pieces of information stored for each instance of a monitoring context. In addition to its type, we must define how the metric value will be determined, either from an inbound event or from some other source when a trigger occurs:

1. Right-click **CreditCheckMonitorModel MC** and select **New** → **Metric**.
2. In the Create New Metric window (Figure 10-49), for Name, type `CustomerNumber`, and for Type, select **Integer**. Click **OK**.

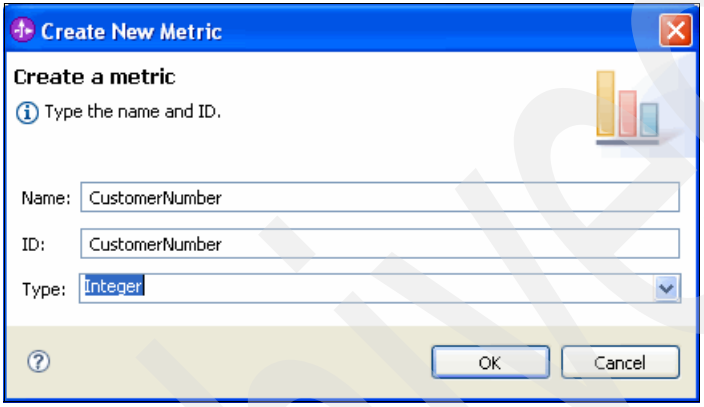


Figure 10-49 Creating a new metric

3. Scroll down to the Metric Value Expressions section (Figure 10-50):
   a. Click **Add** to create a new row in the expression table.

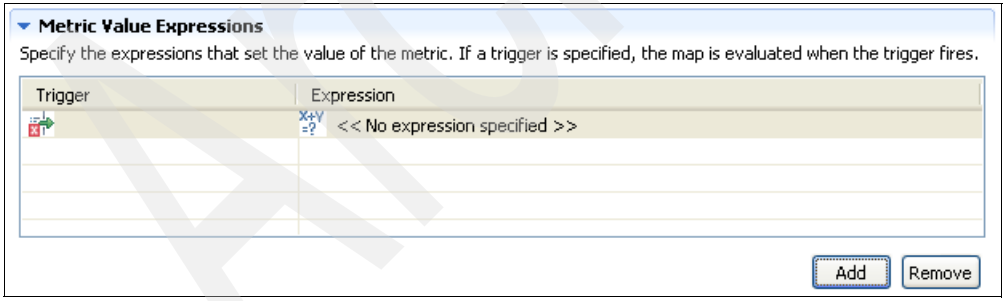


Figure 10-50 Metric Value Expressions section

b. Click in the Expression column and press Ctrl+Spacebar to be prompted with an expression builder, as shown in Figure 10-51. Expand **CreditCheckMonitorModel** → **CreditCheckMonitorModel MC** → **CreditCheckRequestStart** → **Extended Data** and select **CustomerNumber**.

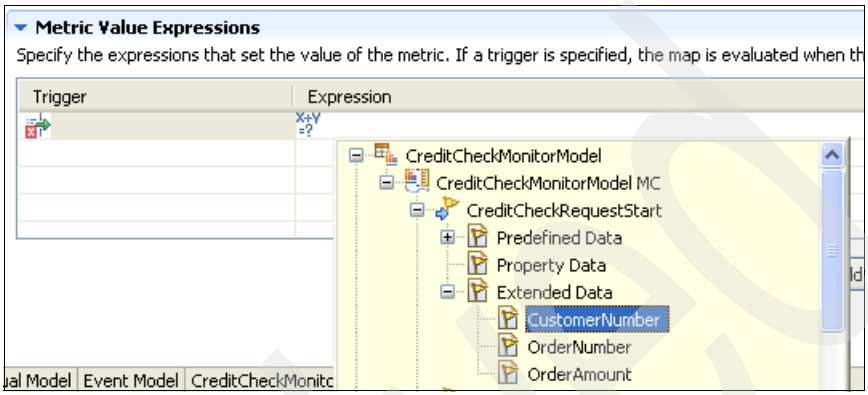


Figure 10-51 Metric Value Expressions builder

This yields the expression shown in Figure 10-52.

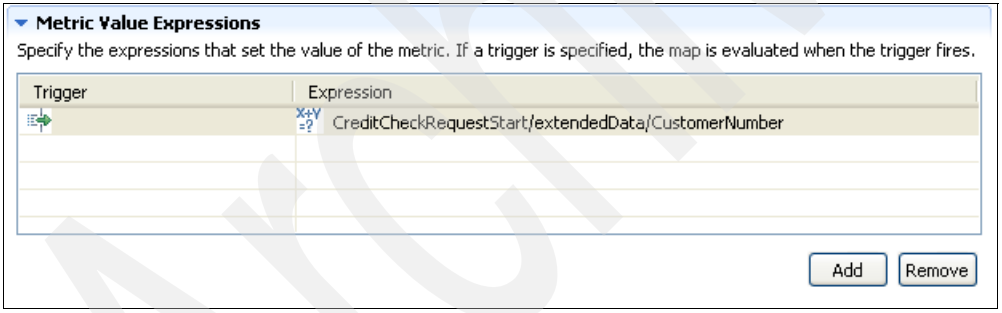


Figure 10-52 Result of adding a metric value expression

4. Repeat the process to create a metric named OrderAmount:

   a. In the Create New Metric window (Figure 10-53), for Name, type `OrderAmount`, and for Type, click **Decimal**.

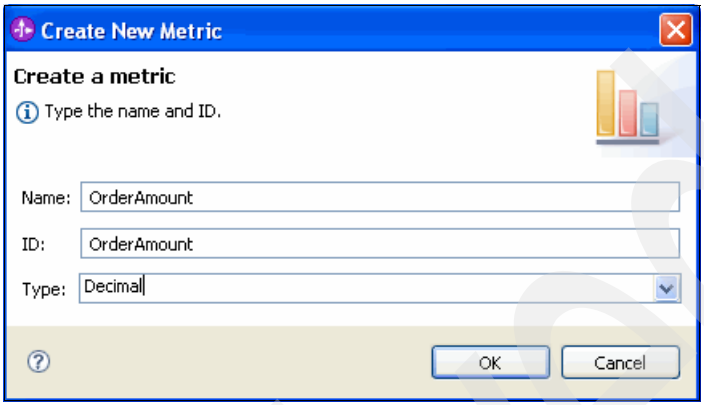


Figure 10-53 Create New Metric window

   b. Build a metric value expression as shown in Figure 10-54. The expression contains the text `CreditCheckRequestStart/extendedData/OrderAmount`.

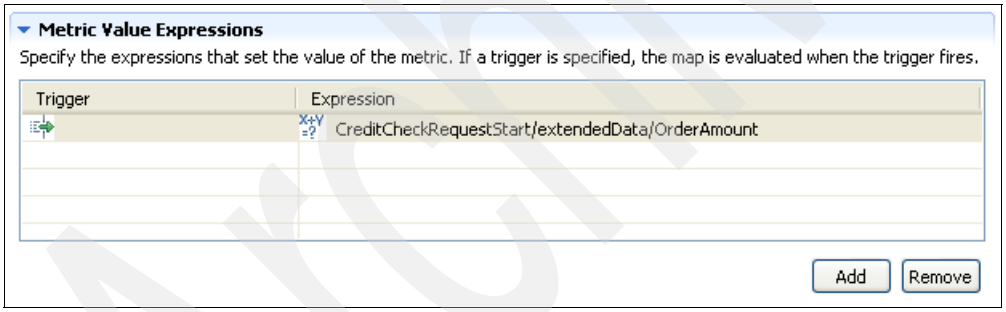


Figure 10-54 Metric Value Expressions result

Archived

5. Repeat the process for the process start time metric:

   a. In the Create New Metric window (Figure 10-55), for Name, type `CCRStartTime`, and for Type, select **DateTime**. Click **OK**.

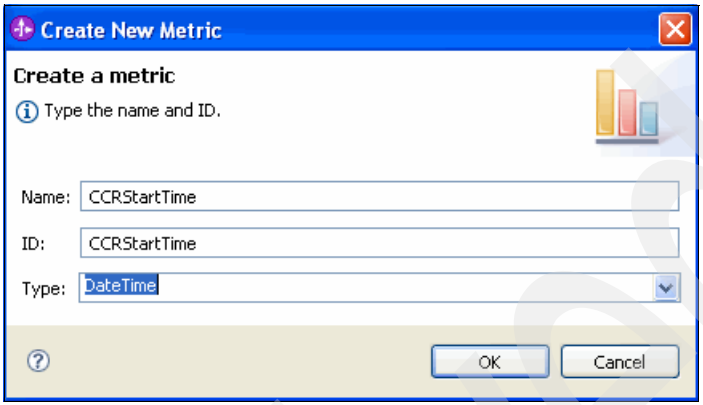


Figure 10-55 Create New Metric window

   b. In the Metric Value Expressions section (Figure 10-56), click in the Expression column and press Ctrl+Spacebar to be prompted with an expression builder. Expand **CreditCheckMonitorModel** → **CreditCheckMonitorModel MC** → **CreditCheckRequestStart** → **Predefined Data** and select **creationTime**.

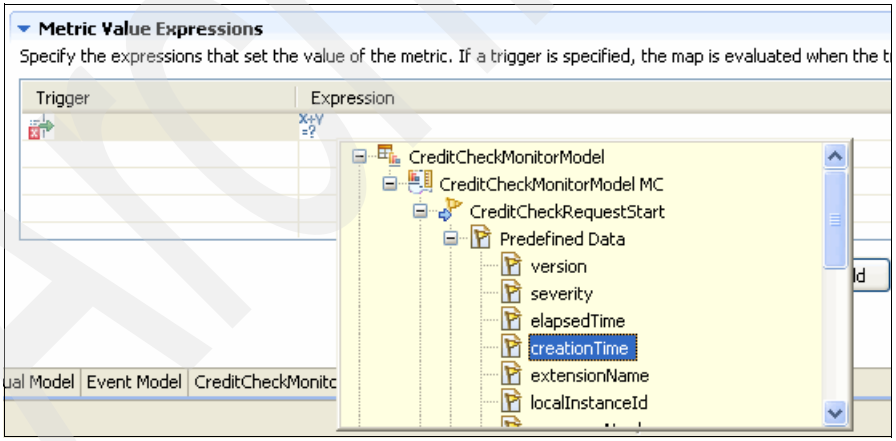


Figure 10-56 Metric Value Expression builder

Figure 10-57 shows the resulting expression with the predefined data.

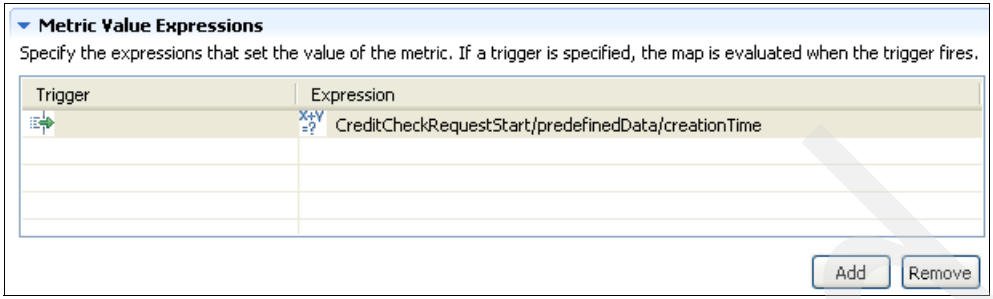


Figure 10-57 Metric Value Expressions result

6. Repeat the process for the process end (or finish) time metric:

   a. In the Create New Metric window (Figure 10-58), for Name, type `CCRFinishTime`, and for Type, select **DateTime**. Click **OK**.

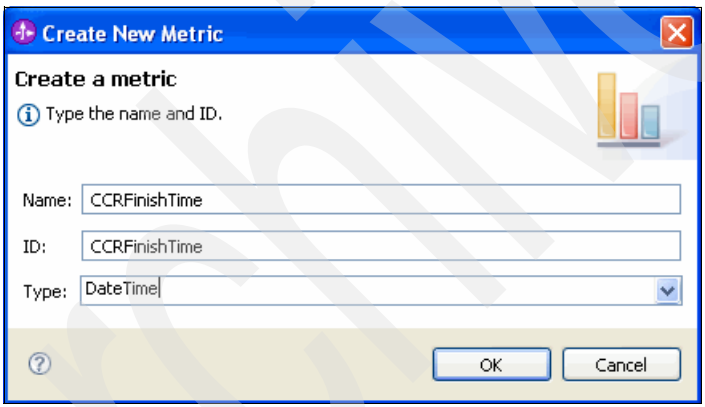


Figure 10-58 Create New Metric window

   b. In the Metric Value Expressions section (Figure 10-59 on page 709), click in the Expression column and press Ctrl+Spacebar to be prompted with an expression builder. Expand **CreditCheckMonitorModel** → **CreditCheckMonitorModel MC** → **CreditCheckRequestFinished** → **Predefined Data** and select **creationTime**.

Archive

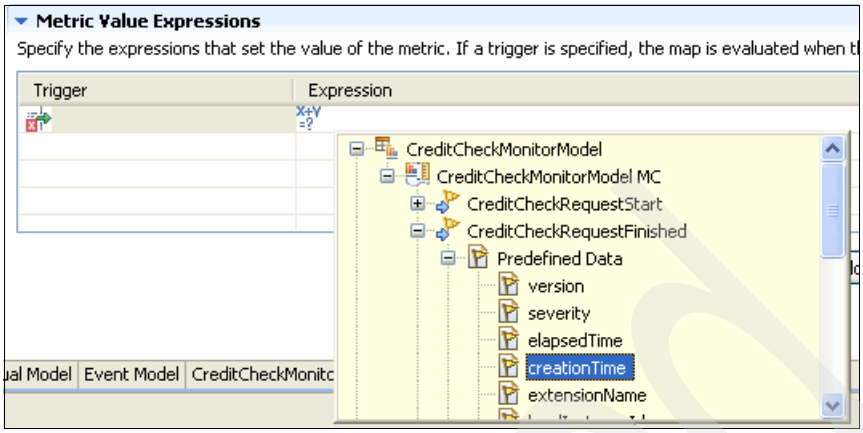


Figure 10-59 Metric Value Expression builder

7. Repeat the process for the duration of the process. For this we need a metric calculated from two other metrics, but only when a certain trigger occurs.

   a. In the Create New Metric window (Figure 10-60), for Name, type `Duration`, and for Type, select **Duration**. Click **OK**.

> **Calculating the duration:** This example uses a simple approach to calculate the duration. Other alternatives might use the stopwatch capability or trigger the calculation based on the setting of a value. Consider different approaches and options for other model designs.

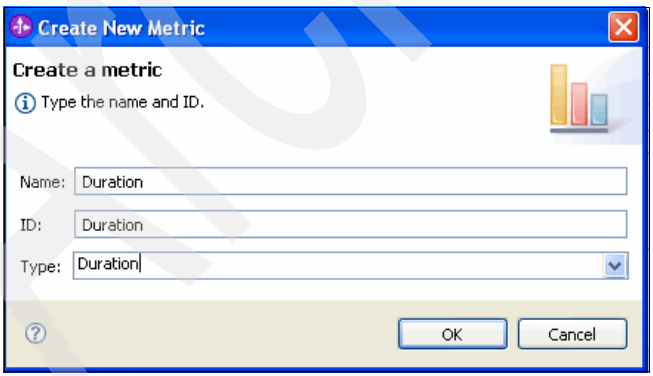


Figure 10-60 Create New Metric window

b. In the Metric Value Expressions section:

i. Click in the Expression column and press Ctrl+Spacebar to be prompted with an expression builder. Expand **CreditCheckMonitorModel → CreditCheckMonitorModel MC** and select **CCRFinishTime**. In the expression line, type a minus (-) sign.

ii. Press Ctrl+Spacebar again to prompt for the expression builder (Figure 10-61). Expand **CreditCheckMonitorModel → CreditCheckMonitorModel MC** and select **CCRStartTime**.

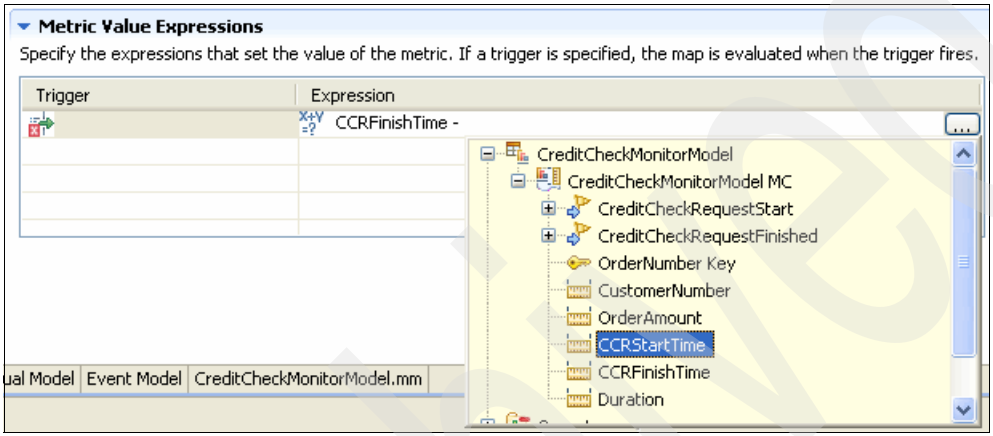


Figure 10-61 Using the Metric Value Expression builder to calculate a value

This provides an expression that calculates the duration from the start and finish time stamps. Figure 10-62 shows the completed calculation.

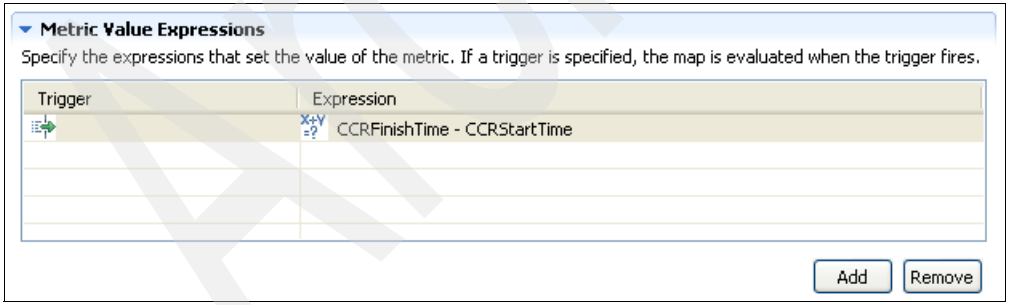


Figure 10-62 Metric Value Expression calculation

c. To make this expression occur, we associate the CreditCheckRequestFinished Trigger with the metric value expression. Double-click in the Trigger column and select the **CreditCheckRequestFinished Trigger**. Figure 10-63 shows the result.

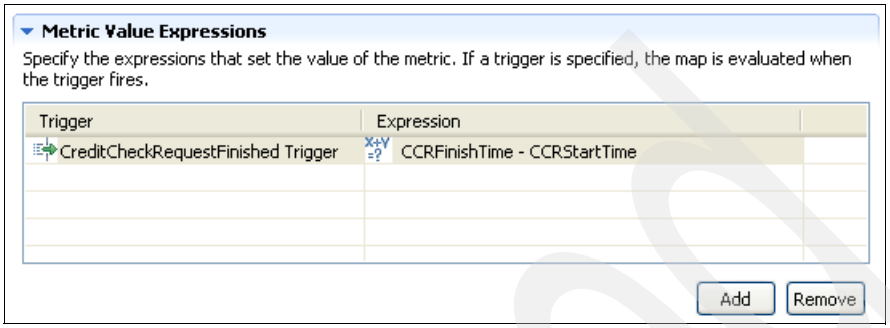


Figure 10-63 Completed Duration metric with a trigger

8. Repeat the process one more time for OrderStatus:

   a. In the Create New Metric window (Figure 10-64), for Name, type `OrderStatus`, and for Type, select **String**. Click **OK**.

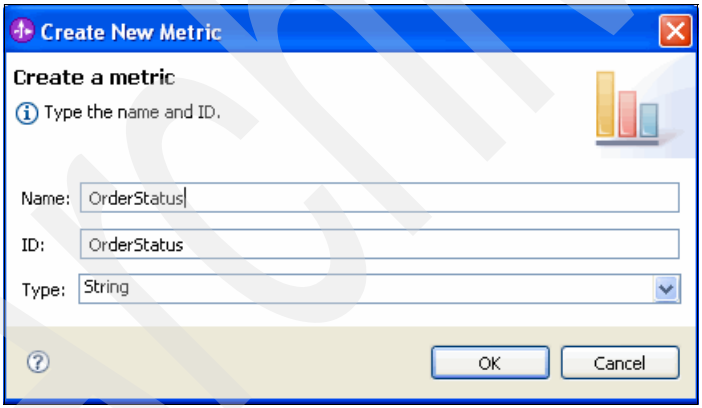


Figure 10-64 Create New Metric window

Archived

b. Build a metric value expression as shown in Figure 10-65. The expression contains the text `CreditCheckRequestFinish/extendedData/OrderStatus`.

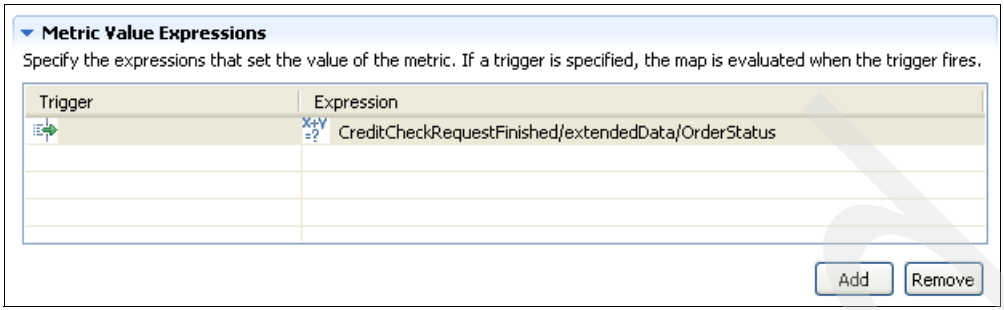


Figure 10-65 Metric Value Expression completed

9. Save the monitor model. No error messages occur based on these steps.

Figure 10-66 shows the contents of the completed Monitor Details Model, with all the metrics defined. This model has seven pieces of information stored for each instance of the monitoring context, one key value and six metrics.

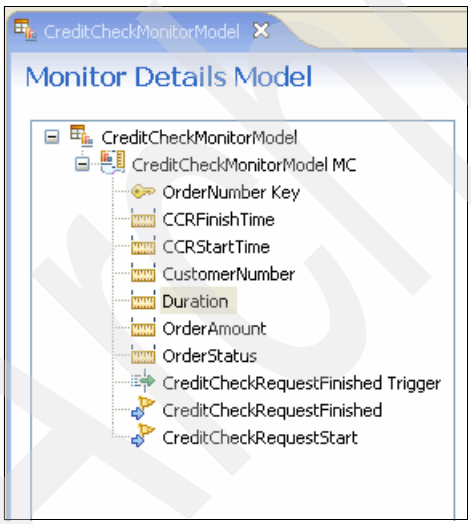


Figure 10-66 Completed monitor model contents

This example represents a full monitoring model. However, for a complete end-to-end working example, you must also define the KPI model.

Archived

Defining the KPI model

In version 6.1 of WebSphere Business Monitor, the business user can create dashboards and KPIs. Still it is likely that some KPIs will be defined within the monitoring model. For this example, we add a KPI named AverageDuration:

1. In the monitoring model editor, click the **KPI Model** tab at the bottom of the panel (Figure 10-67).

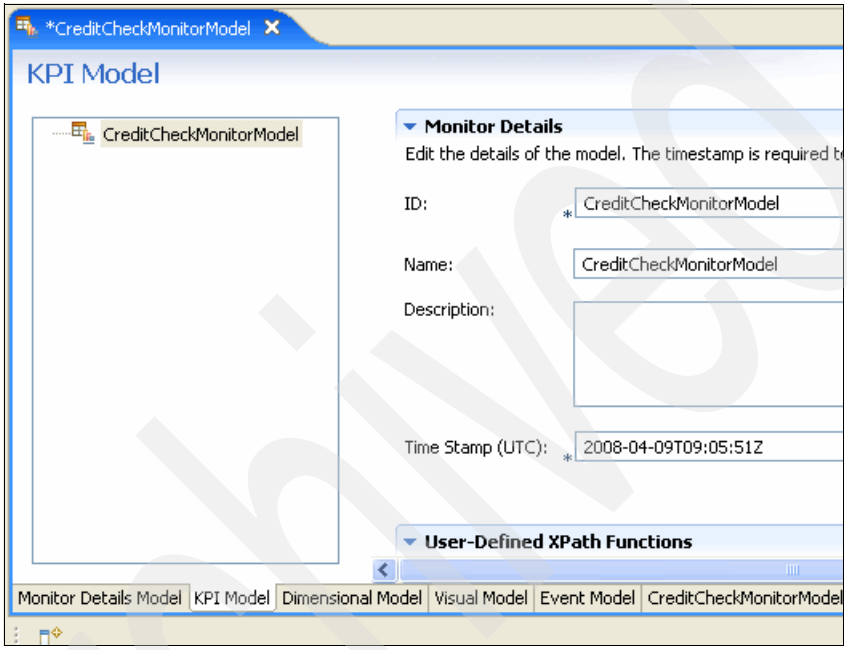


Figure 10-67 KPI Model details panel of the monitor model

2. To create a new KPI context, right-click in the KPI model contents area and select **New** → **KPI Context** (Figure 10-68).

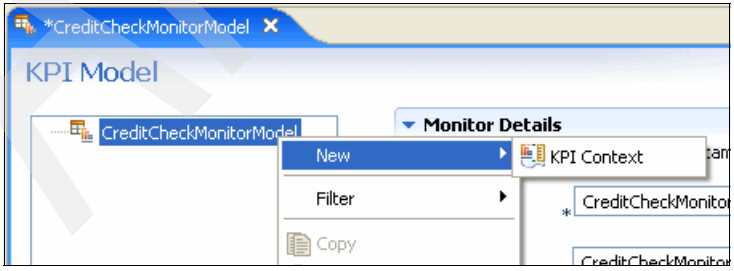


Figure 10-68 Selecting options to create a new KPI context

3. In the Create New KPI Context window (Figure 10-69), for Name, type `CreditCheck KPI Model`. Click **OK**.

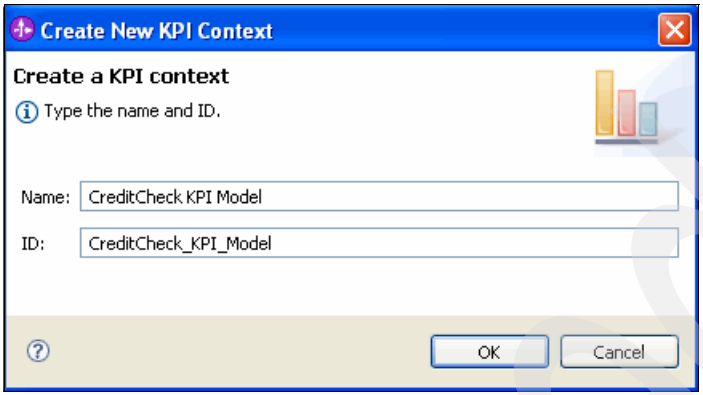


Figure 10-69 Create New KPI Context window

4. Define a new KPI. Right-click the new context, and select **New** → **KPI**.
5. In the Create New KPI window (Figure 10-70), for Name, type `AverageDuration`. Click **OK**.

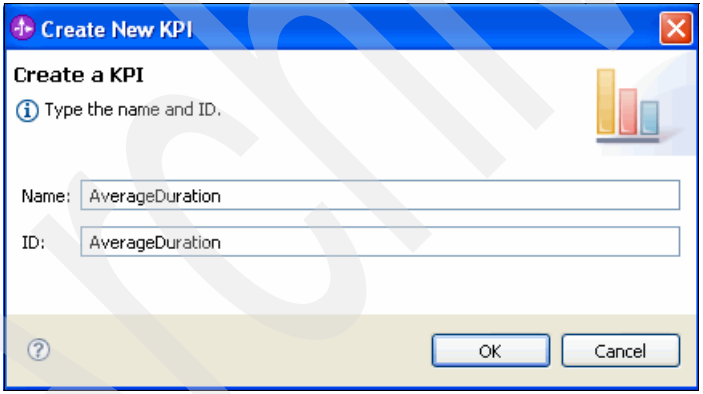


Figure 10-70 Create New KPI window

Archive

6. In the resulting KPI Details section (Figure 10-71), for Type, select **Duration**.

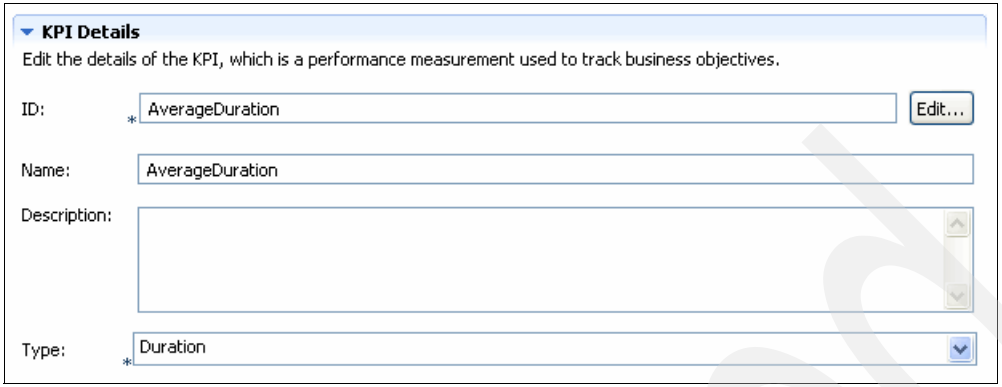


Figure 10-71 KPI Details section

7. Scroll down to the KPI Target and Ranges section:
   a. For Target, click the **Details** button.
   b. In the Target Details window, for Name, type `Target Duration`, and for Target, select **2** for Seconds. Click **OK**.
   c. In the KPI Target and Ranges section, click **Add**.
   d. In the Add Range window, for Name of the first range, type `Low`. Click **OK**.

      Figure 10-72 shows the resulting range definition.

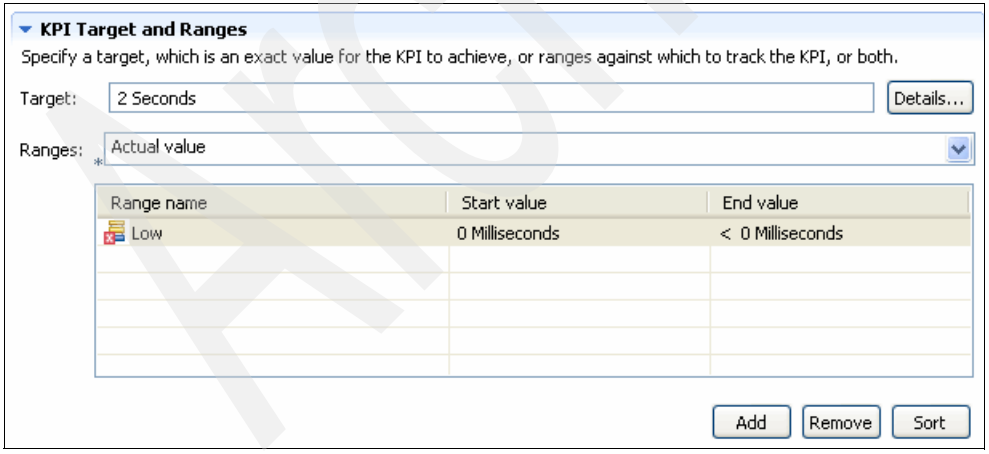


Figure 10-72 KPI Target and Ranges section with Low range added

   e. Click the End value column and click the **edit** button.

Archived

f. In the Specify End Value window (Figure 10-73), for Value, specify **2** for Seconds. Click **OK**.

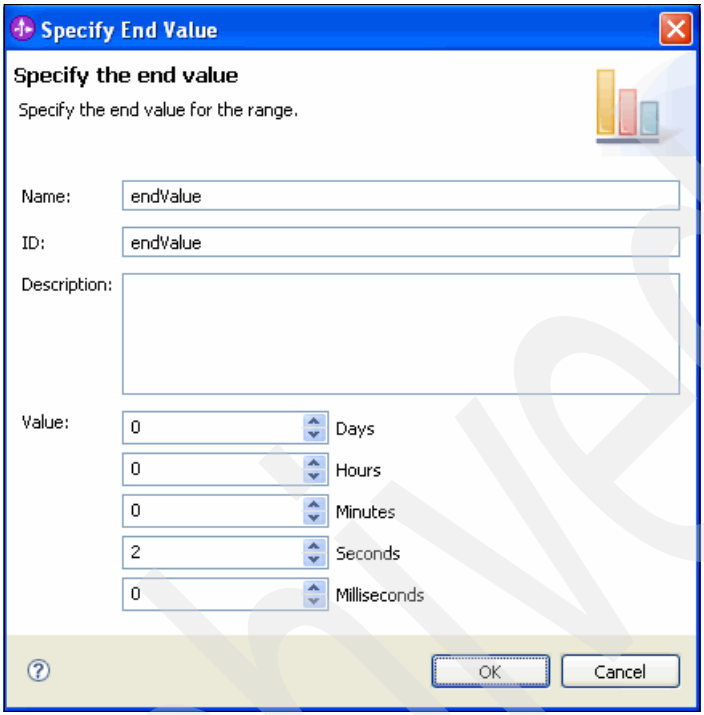


Figure 10-73 Specify End Value window

g. Add a Range called **High** with a Start value of **2** and an End value of **4** seconds.

h. Add a Range called **Critical** with a Start value of **4** seconds and an End value of **5** seconds.

Archived

Figure 10-74 shows the completed section.

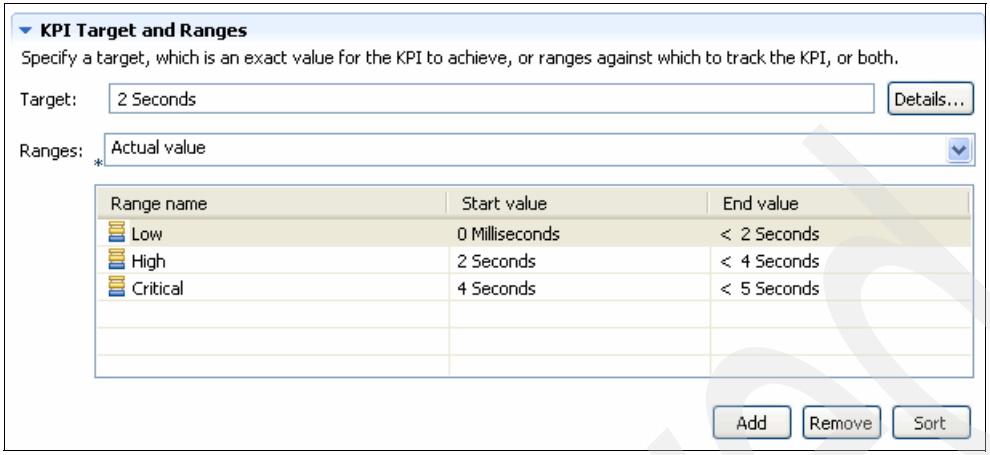


Figure 10-74 KPI Target and Ranges completed

8. Scroll down to the KPI Definition section (Figure 10-75):
   a. Under KPI value, select **Base this KPI on a metric and an aggregation function**.
   b. Under KPI Details:
      i. For Monitoring context, select **Browse...**.
      ii. In the Select Monitoring Context window, select the monitor context **CreditCheckMonitorModel MC** and click **OK** to continue.

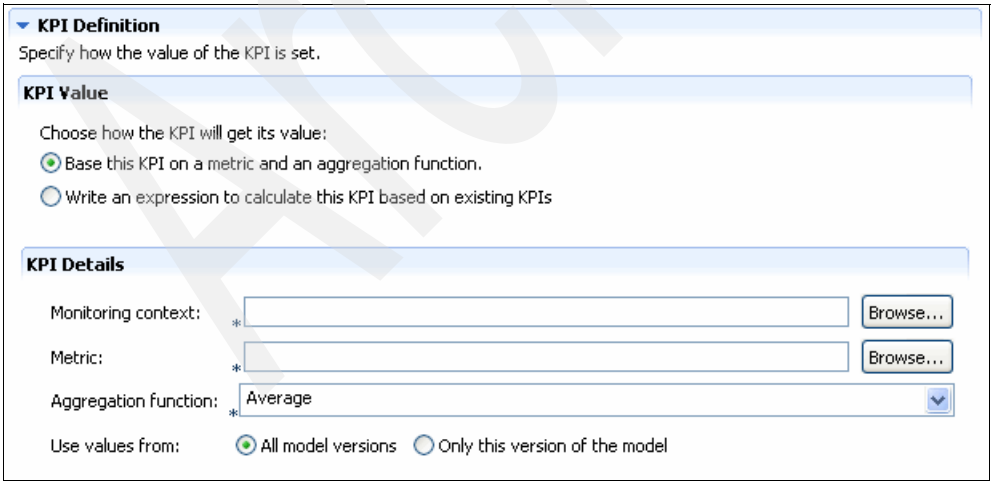


Figure 10-75 KPI Definition Section with KPI Details showing

iii. For the Metric field, select **Browse...**.

iv. In the Select Metric window, select the metric **Duration** and click **OK**.

v. For Aggregate function, select **Average**.

Figure 10-76 shows the results.

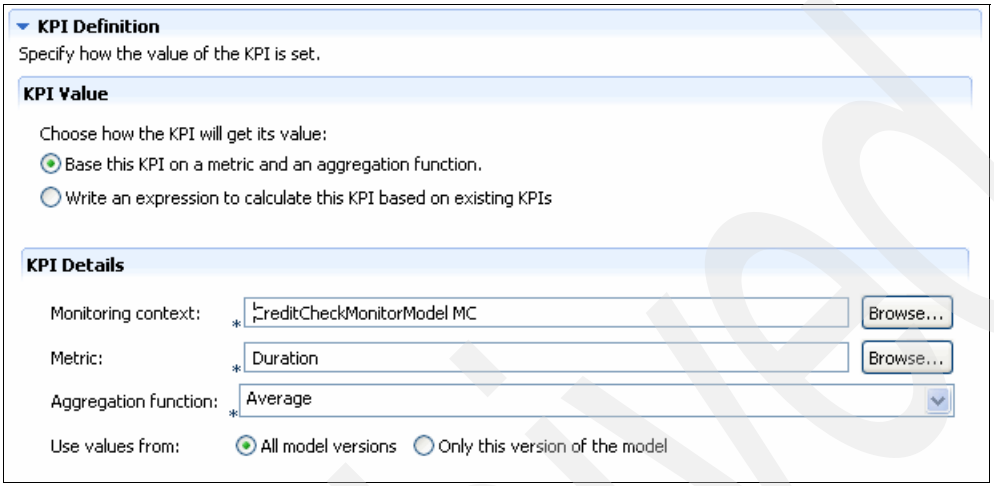


Figure 10-76 KPI Definition with details filled

9. Press Ctrl+S or select **File** → **Save** to save the monitor model.

This completes both the monitor and KPI models.

Archived

10.4.6 Generating the deployment artifacts

In order for the monitor model to operate, it must be packaged as a monitor application and deployed to a WebSphere Business Monitor server:

1. Generate the J2EE components required. In the Project Explorer view (Figure 10-77), right-click **CreditCheckMonitorModel.mm** and select **Generate Monitor J2EE Projects**.

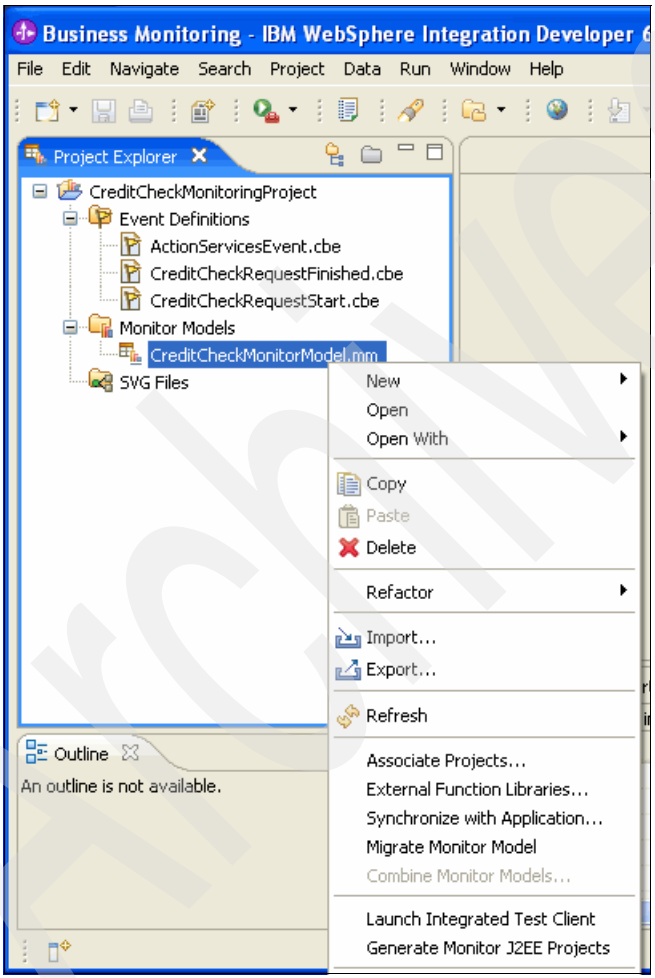


Figure 10-77 Generate Monitor J2EE Projects menu navigation

2. In the Generate Monitor J2EE Projects window, click **Finish** to accept the defaults.

Archived

3. To see the projects that are generated, change to either the Business Integration perspective (Figure 10-78) or the J2EE perspective.

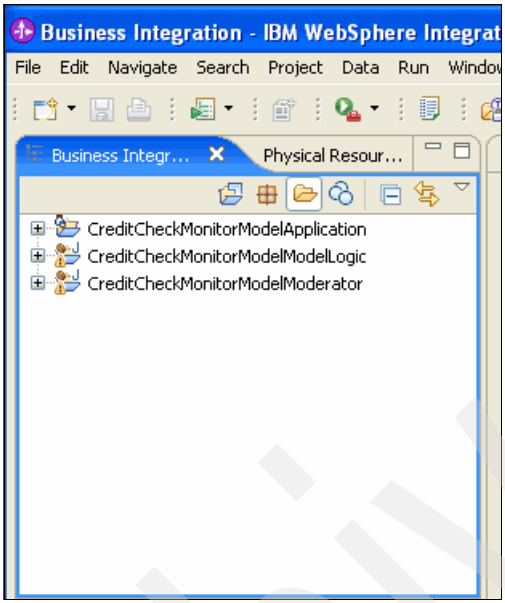


Figure 10-78 Business Integration perspective of the resulting projects

4. To export the required EAR file, right-click the **CreditCheckMonitorModelApplication** project and select **Export** → **EAR file** (Figure 10-79).

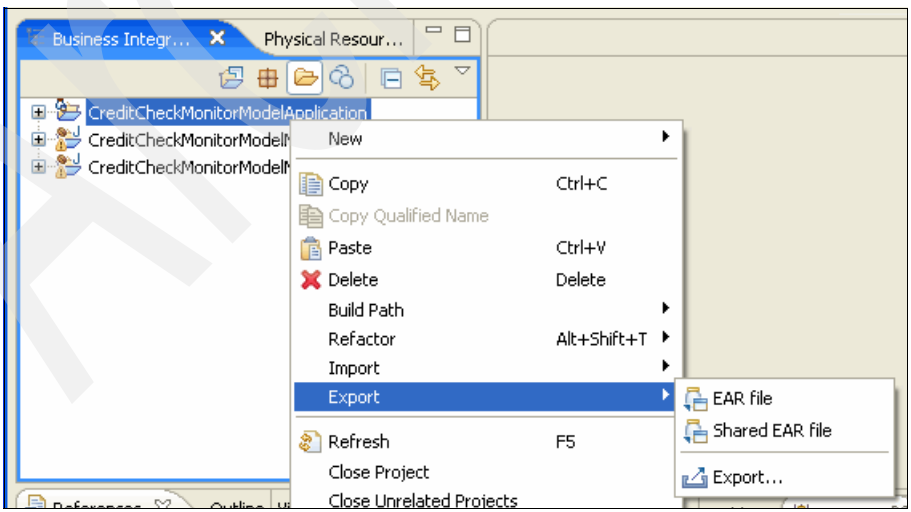


Figure 10-79 J2EE Application export menu navigation

Archived

5. In the Export window (Figure 10-80), for Destination, choose the location for the exported EAR file. For this example, we use the **c:\exports** directory. Click **Finish** to complete the export.

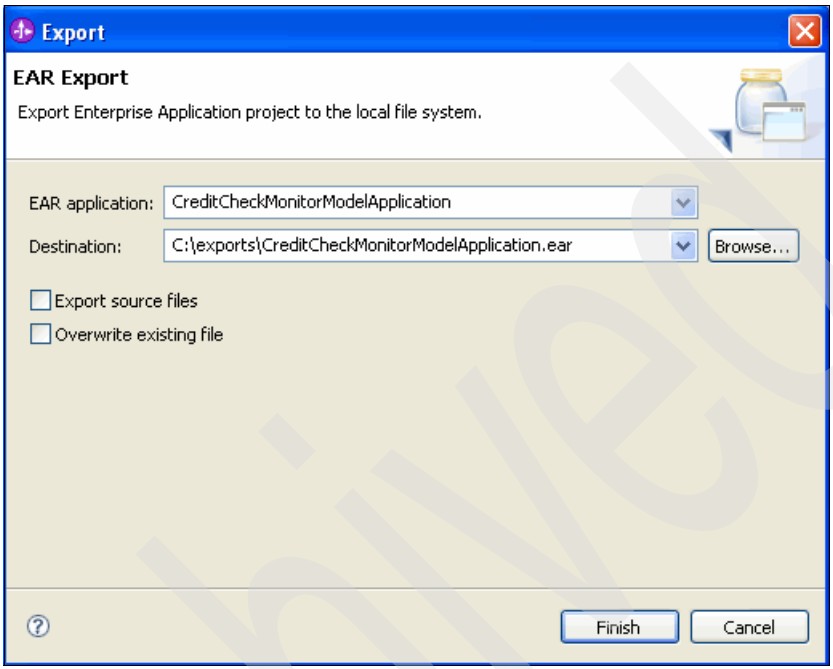


Figure 10-80 Export window

The monitor application EAR file is now ready for deployment.

10.4.7 Deploying the monitor application

By using the administration console of the WebSphere Business Monitor server, install the monitor application:

1. In the administrative console (Figure 10-81 on page 722), expand **Applications** and click **Install New Application**.

Archived

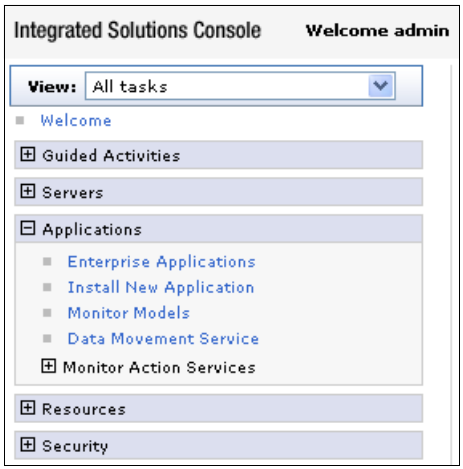


Figure 10-81 Selecting Install New Applications

2. In the Preparing for the application installation panel (Figure 10-82), select **Local File system** as the source, and for Full path, click **Browse**. Navigate to and select the **CreditCheckMonitorModelApplication.ear** file. Click **Next**.

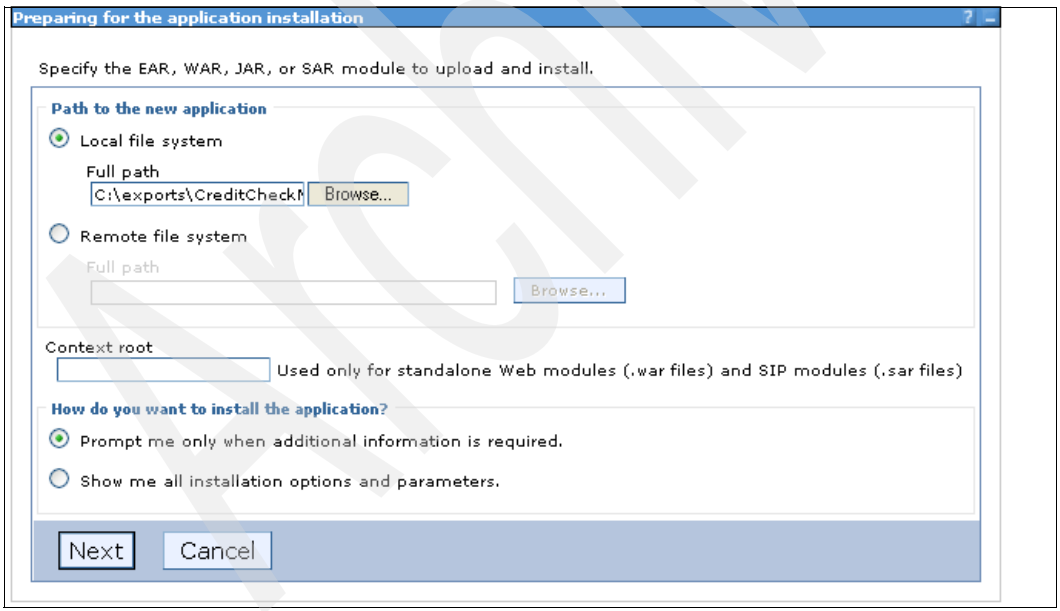


Figure 10-82 Preparing for the application installation panel

3. In the Select installation options panel, review the information and click **Next** to accept the values.

4. In the Map modules to servers panel (Figure 10-83), check that the server and node are the one where the WebSphere Business Monitor is running, and click **Next**.

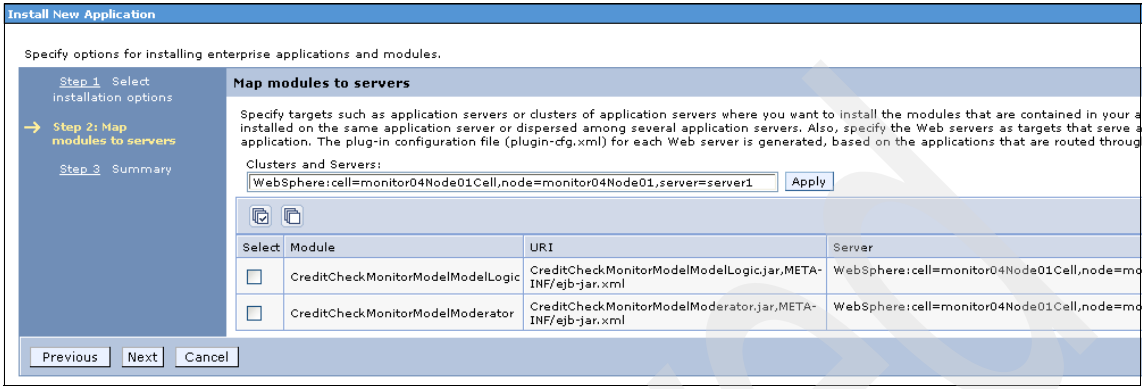


Figure 10-83 Map modules to servers panel

5. In the Summary panel (Figure 10-84), click **Finish** to install the application.

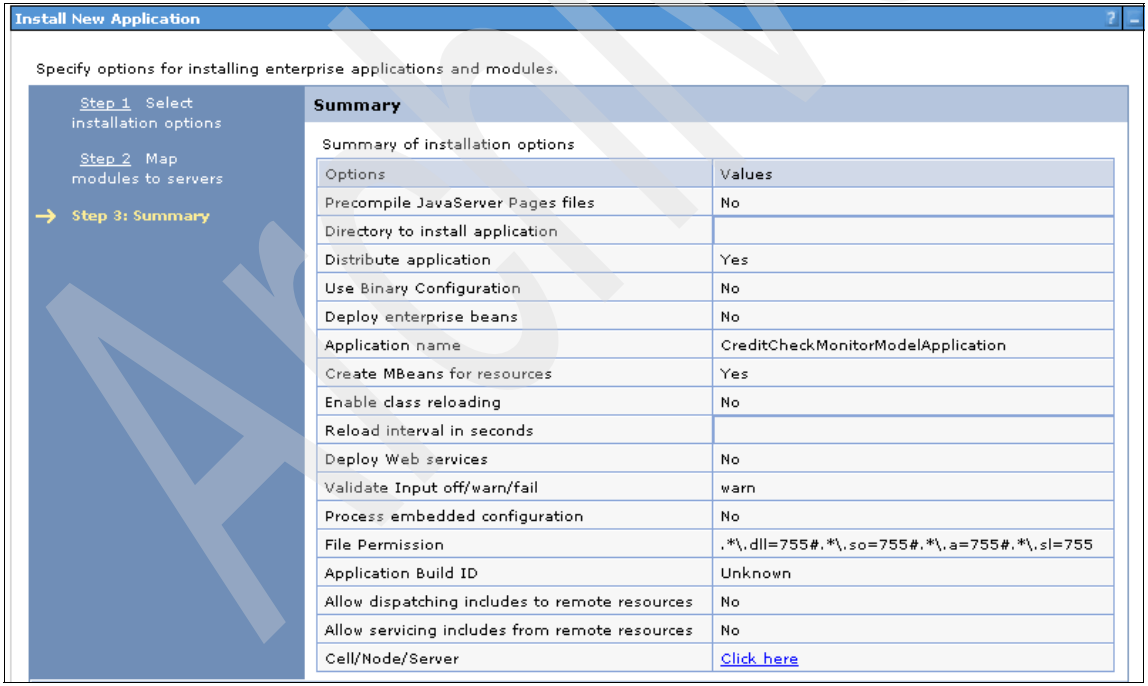


Figure 10-84 Installation summary

6. In the next panel (Figure 10-85), you see the results of the deployment. The results are a list of warning and information messages, the last of which should indicate successful installation. As long as they are all warnings and information messages, then the application is installed properly.

   Click the **Save** link to save the application server master configuration.

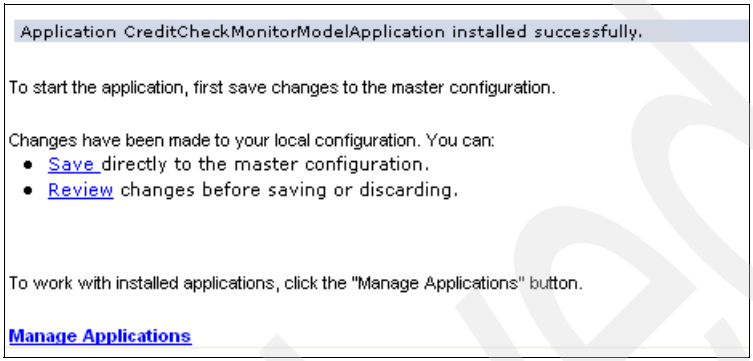


Figure 10-85 End of results panel

7. To verify the installation:

   a. In the navigation pane, expand **Applications** and click **Monitor Models**.

   b. In the Monitor Models panel (Figure 10-86), the Deployment is OK and the Status has a green arrow. (When a model is first deployed, it may take some time before you see the green arrow.) If the model is not started, select **CreditCheckMonitorModel** and click **Start**.

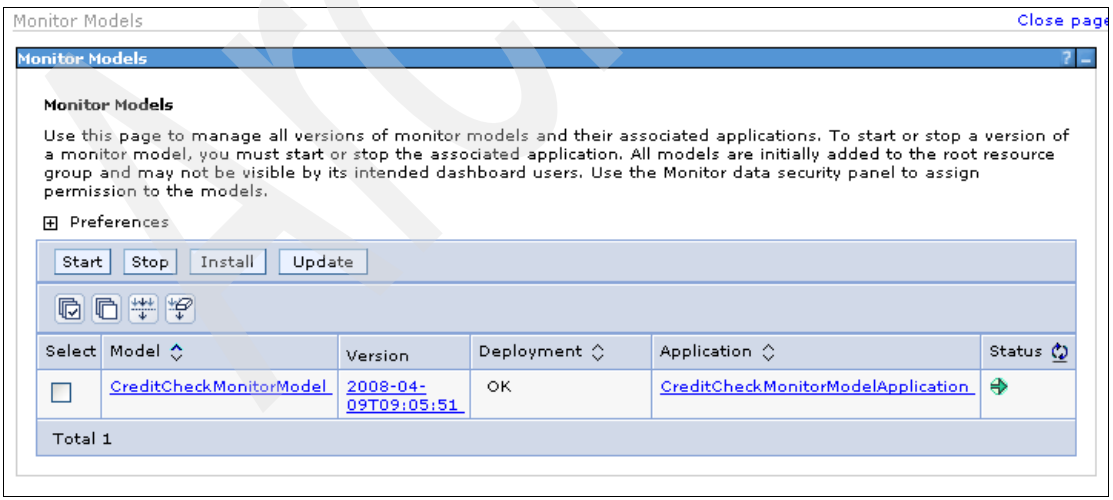


Figure 10-86 Installed monitor models

This completes the installation of the monitor model to the WebSphere Business Monitor server.

10.4.8 Defining the business dashboards

To use the KPI defined, you must create and select a dashboard to display it:

1. By using a browser, connect to the WebSphere Business Monitor dashboard application. For this example, with the monitor installed on machine *monitor04*, we enter the following URL:

   `http://monitor04:9080/BusinessDashboard`

2. Enter a user ID. When security is turned off, any ID is acceptable. In this example, we type `BusinessUser`. Click **Login**.

3. In the Welcome panel (Figure 10-87), click the **Dashboards** tab at the top. Click **New** to define a dashboard.

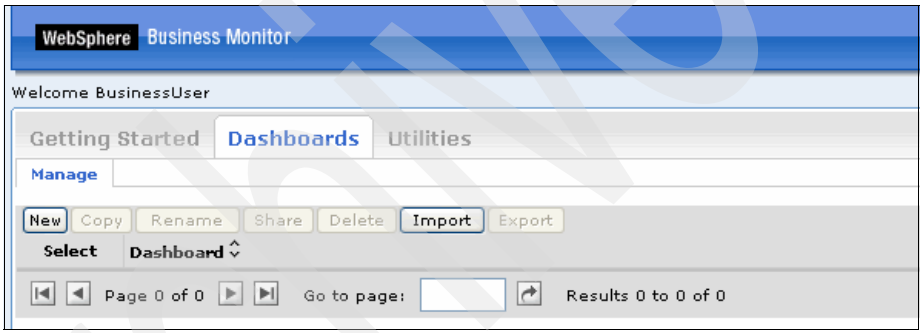


Figure 10-87 Dashboards panel in WebSphere Business Monitor

4. In the New Dashboard panel (Figure 10-88), for Name, type `CC Dashboard` to create a blank dashboard. Click **OK**.

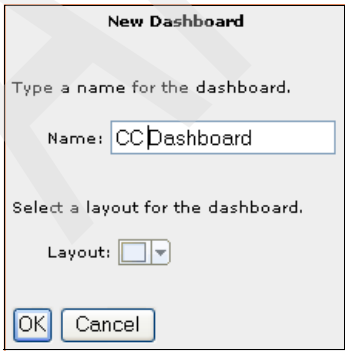


Figure 10-88 New Dashboard panel

Archived

5. In the new dashboard, click the **Add to Dashboard** link.
6. In the Add to Dashboard panel (Figure 10-89), select **KPIs** and click **OK**.

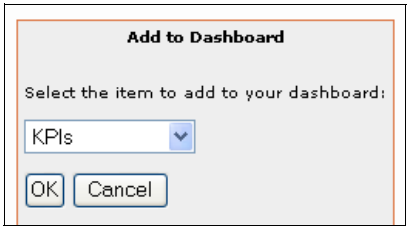


Figure 10-89 Add to Dashboard panel

7. In the CC Dashboard panel, click **Personalize** to select which KPI to display.
8. In the KPIs - Select KPIs panel, select **AverageDuration**.
9. Click the **Layout** tab.
10. In the KPIs - Layout panel (Figure 10-90), select **Half Gauge** with a **Horizontal** layout. Accept the defaults for the other values. Click **OK**.

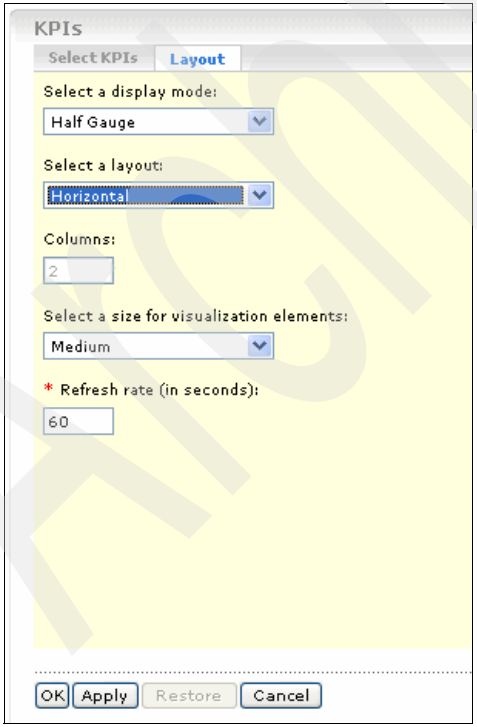


Figure 10-90 Selecting a half gauge to display the predefined KPI

Archived

This adds a graphical gauge to the dashboard (Figure 10-91).

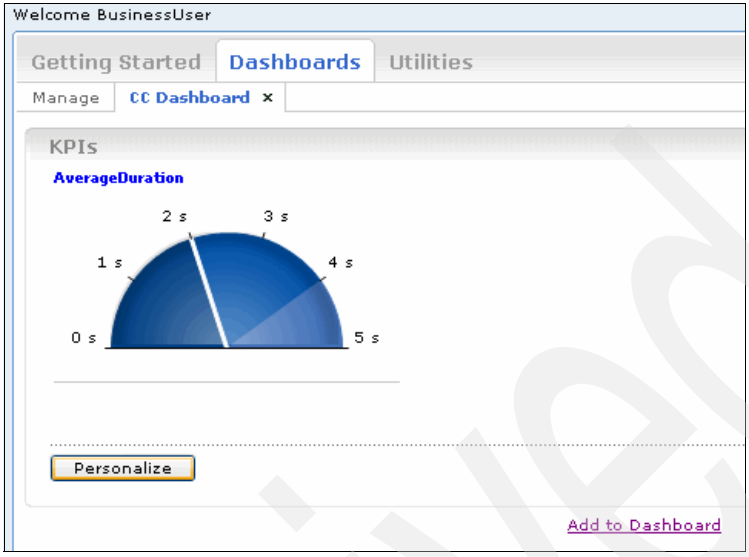


Figure 10-91 AverageDuration KPI with half gauge display

11. For testing purposes, add another component to the dashboard:
    a. Click **Add to Dashboard**.
    b. In the Add to Dashboard panel, select **Instances**. Click **OK** to add an Instances panel to the dashboard.
12. In From the Instances panel, click **Personalize**.

Archived

13. In the Instances - Show/Hide window (Figure 10-92), select the metrics to display in the Instances panel so that we can see the events that are being handled. For this example, we choose the following columns:

    – CustomerNumber
    – OrderNumber Key
    – OrderAmount
    – OrderStatus
    – Duration

    Click **OK**.

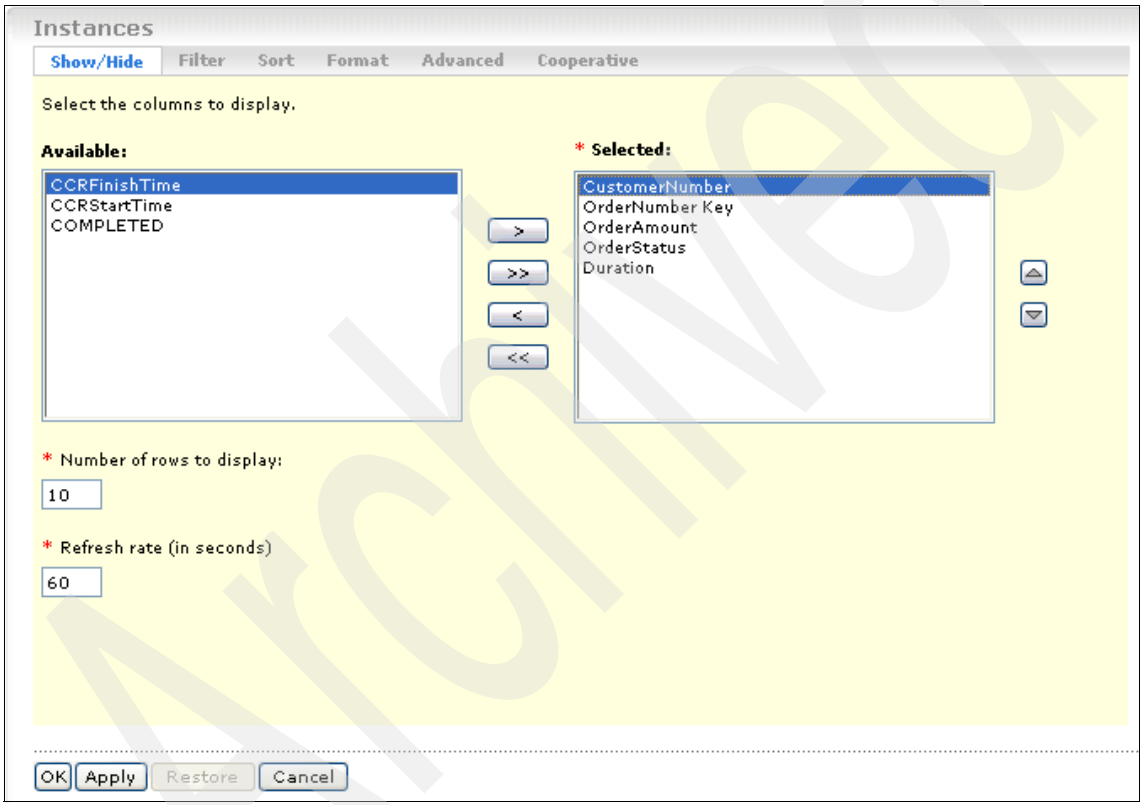


Figure 10-92 Selecting the columns to display in the dashboard

Archived

Figure 10-93 shows the resulting dashboard.

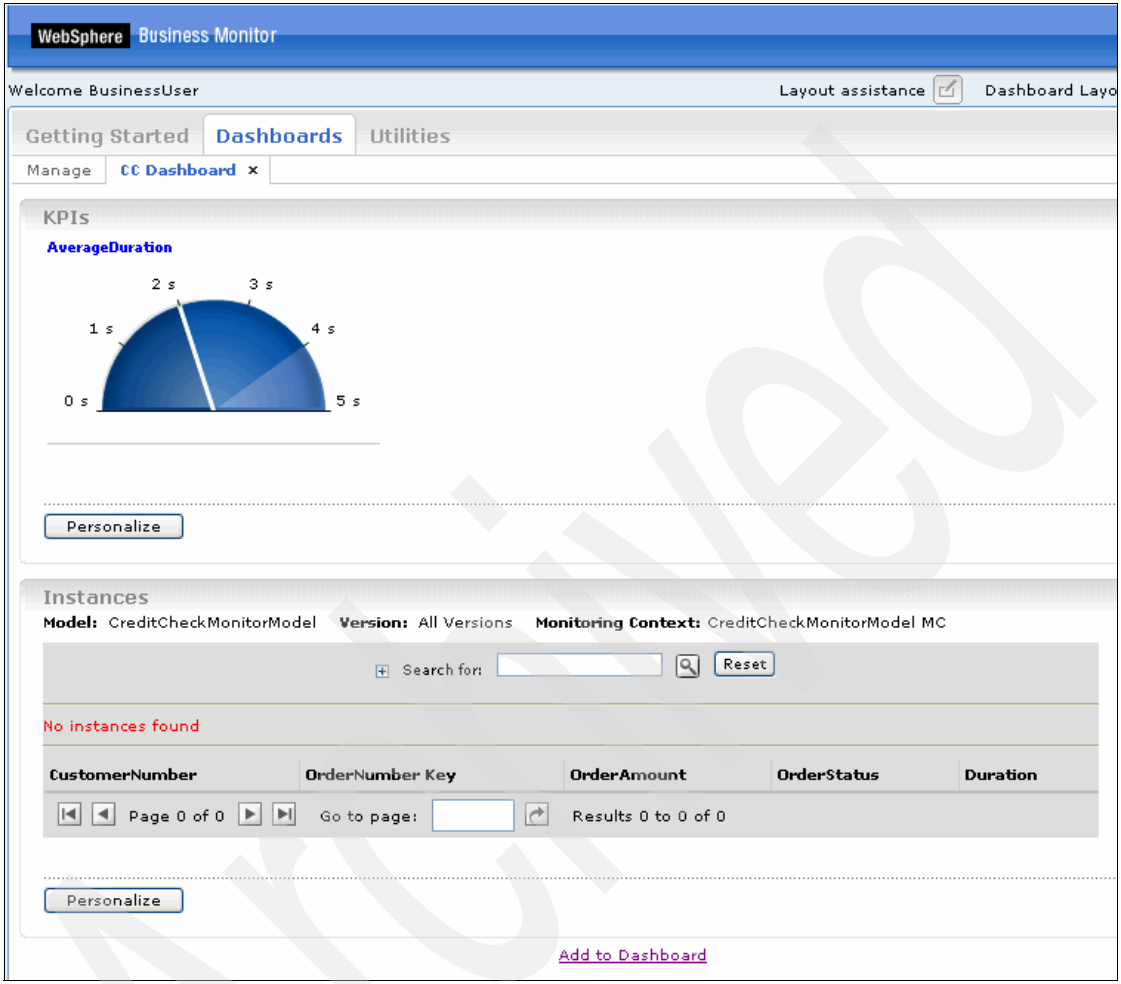


Figure 10-93 Finished dashboard

These steps yield an operational dashboard for the Credit Check application. You are free to add KPIs and other dashboard components as needed, but the only data available is captured in the defined monitor model.

10.4.9 Testing by using ad hoc event generation

In the additional material for this book (Appendix B, “Additional material” on page 801), a simple Java application is provided to generate test Common Base Events for this example. The events that are generated are modeled as those

that are generated from the CreditCheckRequest application. Each time the sample program is run, ten pairs of events (a start and finished event pair) are generated and placed on the WebSphere MQ queue CEIMQ. The program prints the Common Base Events to standard out (the console) for reference.

The code is provided as a project interchange file for loading into a WebSphere Integration Developer or Rational Application Developer development environment. The Java Naming and Directory Interface (JNDI) definitions for the connection factories and the queue are in the JNDI folder of the project. WebSphere MQ Explorer was used to create these definitions and can be used to modify them as needed.

To load and run the program:

1. Start WebSphere Integration Developer and create a new workspace.
2. In the Import Project Interchange Contents window (Figure 10-94), import the provided project interchange file, CreditCheckTest001.zip. Click **Finish**.

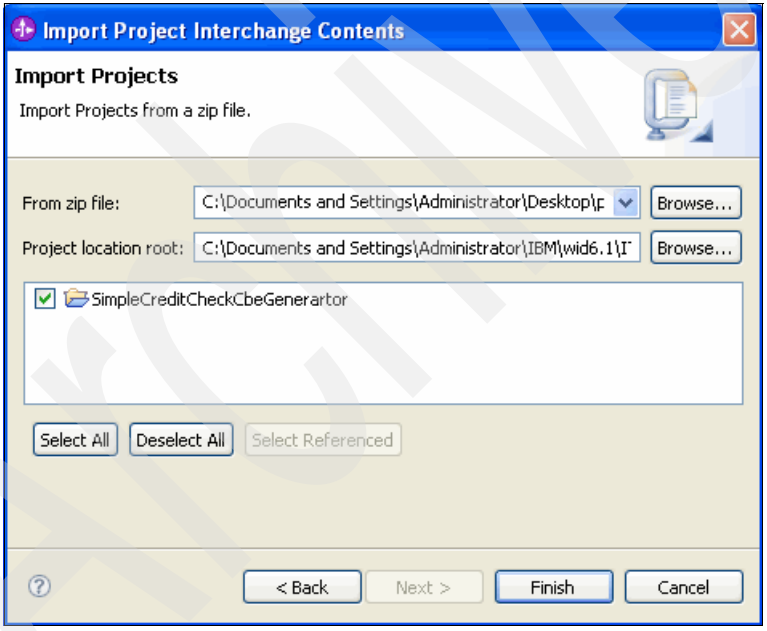


Figure 10-94 Import Project Interchange Contents window

3. In the Business Integration view (Figure 10-95), expand and navigate the classes to **com.ibm.itso.redbook.SimpleCreditCheckCbeGenerator.java**.

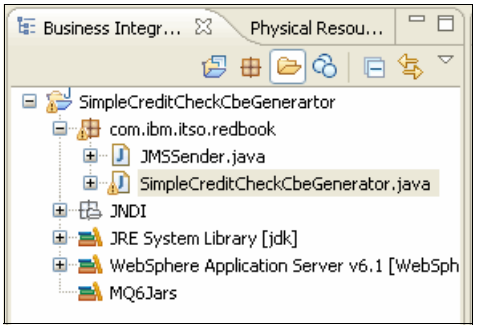


Figure 10-95 Project contents

4. Right-click **SimpleCreditCheckCbeGenerator.java** and select **Run As** → **Java Application** to generate sample events through WebSphere MQ to the CEI and the deployed monitoring model.

Note: The JNDI definitions provided expect to reach the MQ machine with the host name mq01. If a different name is used, then adjust for the situation by one of the following options:

- Add mq01 as a host name, with its IP address, in the local machine's host file.
- Search for mq01 in the JNDI or .bindings file and change to the host name of the machine of the queue manager.
- Use WebSphere MQ Explorer to open and modify the connection factory defined in the JNDI or .bindings file.

Archived

Figure 10-96 shows a typical CreditCheckRequestStart Common Base Event generated by this program. Notice the extensionName of CreditCheckRequestStart. It is important that this value be present. The monitoring model is set to expect this.

```
<CommonBaseEvent creationTime="2008-04-14T20:01:12.999Z"
extensionName="CreditCheckRequestStart"
globalInstanceId="CEA1DD0A5D888A6370FF358CC22DBFF4F1" msg="ITSO BAM
Event Test" version="1.0.1">
   <extendedDataElements name="OrderNumber" type="int">
      <values>1317405254</values>
   </extendedDataElements>
   <extendedDataElements name="CustomerNumber" type="int">
      <values>57012</values>
   </extendedDataElements>
   <extendedDataElements name="OrderAmount" type="double">
      <values>423.0</values>
   </extendedDataElements>
   <sourceComponentId application="Windows" component="svchost.exe"
componentIdType="win386_svc" location="9.45.72.138"
locationType="IPV4" subComponent="tlntsvr.exe"
componentType="http://www.ibm.com/namespaces/autonomic/Windows"/>
   <situation categoryName="ReportSituation">
      <situationType
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ReportSituation" reasoningScope="INTERNAL"
reportCategory="LOG"/>
   </situation>
</CommonBaseEvent>
```

Figure 10-96 Sample CreditCheckRequestStart Common Base Event

Archived

Figure 10-97 shows a typical CreditCheckRequestFinished Common Base Event generated by this program. Notice the extensionName of CreditCheckRequestFinished. It is important that this value be present. The monitoring model is set to expect this.

```
<CommonBaseEvent creationTime="2008-04-14T20:01:14.953Z"
extensionName="CreditCheckRequestFinished"
globalInstanceId="CEA1DD0A5D89B48B90FF358CC22DBFF4F1" msg="ITSO BAM
Event Test" version="1.0.1">
   <extendedDataElements name="OrderNumber" type="int">
      <values>1317405254</values>
   </extendedDataElements>
   <extendedDataElements name="OrderStatus" type="string">
      <values>Approved</values>
   </extendedDataElements>
   <sourceComponentId application="Windows" component="svchost.exe"
componentIdType="win386_svc" location="9.45.72.138"
locationType="IPV4" subComponent="tlntsvr.exe"
componentType="http://www.ibm.com/namespaces/autonomic/Windows"/>
   <situation categoryName="ReportSituation">
      <situationType
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ReportSituation" reasoningScope="INTERNAL"
reportCategory="LOG"/>
   </situation>
</CommonBaseEvent>
```

Figure 10-97 CreditCheckRequestFinished Common Base Event

Archived

The monitor dashboard is displayed as shown in Figure 10-98.

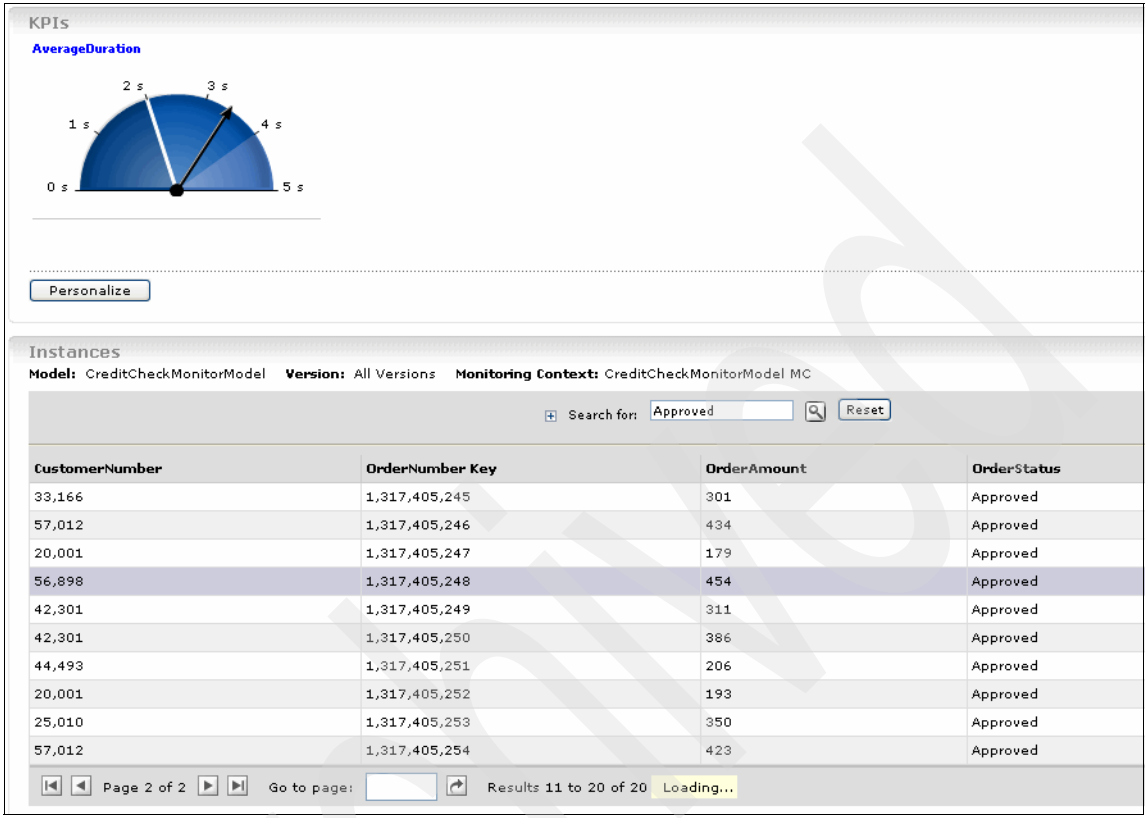


Figure 10-98 Typical results in the dashboard following the test run

In this section, we loaded a test program into the development environment and used it to generate sample events to WebSphere MQ for the monitor model. We viewed the events and saw them reflected in the defined dashboard.

10.5 Summary

In this chapter, we described how to integrate WebSphere MQ with WebSphere Business Monitor through the CEI. We explained the steps to create a basic queue manager, bridge the products, and then create a monitor model from scratch. We also demonstrated the operations by sending test events and seeing the results reflected in the defined dashboard.

Part 3

Appendixes

In this part, we provide additional information that supports the contents of this book:

- Appendix A, “Security activation for monitoring WebSphere Process Server human tasks” on page 737

  In this appendix, we demonstrate how to enable security in WebSphere Process Server in order to monitor human tasks.

- Appendix B, “Additional material” on page 801

  In this appendix, we explain how to obtain the additional material for this book. The additional material is used to help implement the business activity monitoring (BAM) solutions in Part 2, “Business activity monitoring” on page 137.

© Copyright IBM Corp. 2008. All rights reserved.

Archived

A

Security activation for monitoring WebSphere Process Server human tasks

This appendix serves as an extension of Chapter 5, “WebSphere Process Server” on page 149. In this appendix, we provide information about the steps required to enable monitoring of WebSphere Process Server human tasks.

© Copyright IBM Corp. 2008. All rights reserved.

Securing WebSphere Process Server and WebSphere Business Monitor

The monitoring of human tasks requires an extension to a WebSphere Process Server – WebSphere Business Monitor environment by a monitor model that can handle events from human tasks. Furthermore, it requires security enabled on both WebSphere Process Server and WebSphere Business Monitor in order to provide a reliable way to associate a task to a person.

We recommend that both servers use the same user registry. In this scenario, a Lightweight Directory Access Protocol (LDAP) registry in place. In this appendix, we assume that such a registry is already set up. Table A-1 shows all LDAP information required for the security setup and the values used in our environment.

Table A-1 LDAP properties

Property	Parameter
LDAPHostName	9.42.170.182
LDAPPort	389
LDAPServerType	IBM Tivoli Directory Server V6.0
LDAPBindID	cn=root
LDAPBindPassword	detlab0
LDAPSuffix	o=customer

The LDAP Directory Interchange Format (LDIF) file (orig.ldif) used in this appendix is provided in the additional material for this book. You must import the fie into the LDAP server. Figure A-1 on page 739 shows the user structure.

Note: The *wsadmin* user is used as the WebSphere administrative user. The password for this user according the LDIF file is *itso4you*.

Archived

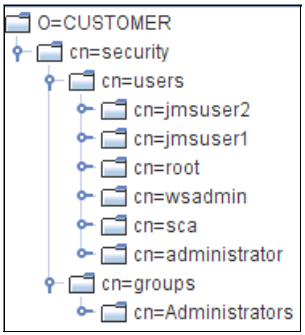


Figure A-1 LDAP user structure

In this section, we explain how to enable WebSphere Process Server human task monitoring for the environment build up in Chapter 5, “WebSphere Process Server” on page 149. This is a WebSphere Business Monitor – WebSphere Process Server environment using a remote Common Event Infrastructure (CEI) hosted by WebSphere Process Server. Both servers currently have security disabled and the service integration bus is set up.

To enable monitoring of WebSphere Process Server human tasks, you must perform the following steps, which are explained further in the sections that follow:

1. Enable event emitting for WebSphere Process Server human tasks.
2. Install the Global Human Task monitor model.
3. Connect WebSphere Process Choreographer with the WebSphere Business Monitor Dashboard.
4. Enable security for WebSphere Business Monitor.
5. Enable security for WebSphere Business Monitor applications.
6. Update the Alphablox Server.properties file.
7. Update J2C authentication data entries for messaging buses for WebSphere Business Monitor.
8. Enable security for WebSphere Business Monitor resources.
9. Modify Alert User IDs.
10. Enable data security for monitor models.
11. Enable security for WebSphere Process Server
12. Enable security for WebSphere Process Server applications.
13. Update the J2C authentication data entries for messaging buses for WebSphere Process Server.

14. Grant access to the foreign queues of the monitor models.
15. Configure the server-to-server Secure Sockets Layer (SSL) from WebSphere Business Monitor to WebSphere Process Server.
16. Configure the dashboard for human task monitoring.

For more information about human task monitoring and security configuration, refer to the following topics in the WebSphere information center:

- Configuring human task monitoring

  http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.btools.help.monitor61.doc/admin/cfg_htm.html

- Configuring security for human task monitoring

  http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.btools.help.monitor61.doc/security/cfg_sec_htm.html

Enabling event emitting for WebSphere Process Server human tasks

The Global Human Task monitor model requires a set of events that are emitted by human tasks at run time. In this section, we explain how to enable event emitting for human tasks in the ClipsAndTacks order handling process:

1. Open the ClipsAndTacks workspace in WebSphere Integration Developer.
2. In the Business Integration perspective, open the ClipsAndTacks order handling process under **ClipsAndTacks** → **Business Logic** → **Processes** → **OrderHandling**. Select the **Review Order** human task.
3. Click the **Properties** tab and click the **Event Monitor** subtab (Figure A-2). Select **All** to emit all human task events at run time.

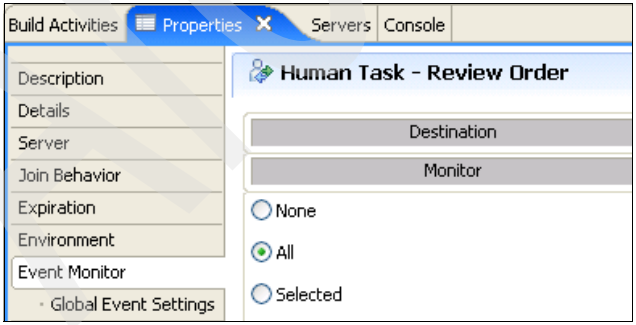


Figure A-2 Enabling event emitting for human tasks

4. Select the **Ship Order** human task.

5. Click the **Properties** tab and the **Event Monitor** subtab. Select **All** to emit all human task events at run time.
6. Press Ctrl+S to save your work.
7. Click the **Servers** tab and select **WebSphere Process Server 6.1**. Click the **Publish to the server** icon to publish your changes to the server (Figure A-3).

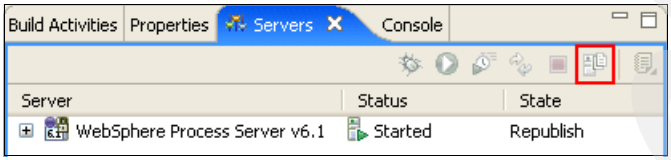


Figure A-3 Republish WebSphere Process Server

Installing the Global Human Task monitor model

Human task monitoring requires a monitor model that is capable of handling incoming human task events. WebSphere Business Monitor provides a Global Human Task monitor model in the installation files. It covers basic human task monitoring capabilities for Business Process Execution Language (BPEL) applications. The metrics provided by this monitor model are unspecific to an application domain.

For more sophisticated, application-specific human task monitoring, IBM provides the Human Task SupportPac for human task monitoring. This SupportPac contains a Project Interchange file that can be extended by using the WebSphere Business Monitor Toolkit.

For further information about the Human Task SupportPac, refer to the following Web address:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.monitor.supportpac.doc/samples/htm_about.html`

In this appendix, we use the Global Human Task monitor model. You can either install this monitor model as part of the WebSphere Business Monitor installation or install it manually afterwards. To install the Global Human Task monitor model:

Archived

1. Verify that the Global Human Task monitor model is already installed:
   a. In the WebSphere Business Monitor administrative console, expand **Applications** and click **Monitor Models**.

      Note: You can access the administrative console of a WebSphere Business Monitor server by using the following URL:

      `http://<host name>:<WC_adminhost port>/ibm/console`

   b. Check whether the list of monitor models contains **GlobalHTMM** (Figure A-4). If it contains the monitor model, follow the steps in “Connecting Business Process Choreographer with a WebSphere Business Monitor dashboard” on page 744. If the monitor model is not installed, continue with the next step.

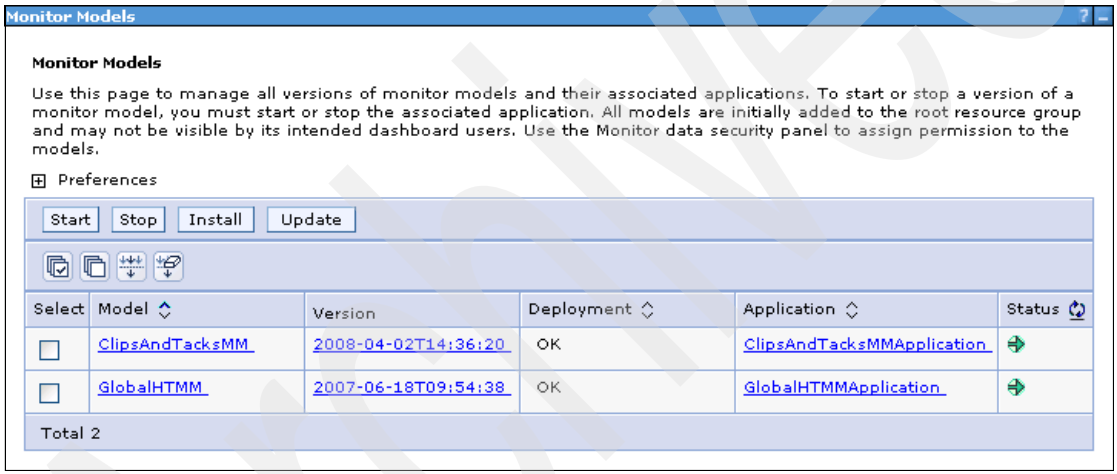


Figure A-4 Installed GlobalHTMM monitor model

2. In the Monitor Models panel (Figure A-5), click **Install**.

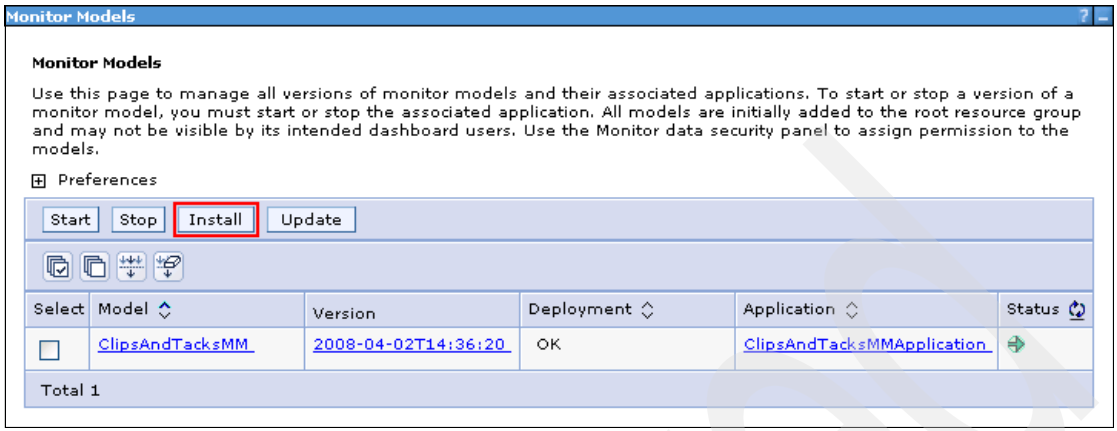


Figure A-5 Installing the GlobalHTMM monitor model

3. Select **Local file system** and click **Browse**. Navigate to the was_root/installableApps.wbm/monitorModels folder that contains the EAR file. Select the **GlobalHTMMApplication.ear** file and click **Open**.

GlobalHTTMMApplication EAR file: The GlobalHTTMMApplication.ear is provided with the WebSphere Business Monitor installation files. Therefore, you must perform these steps on the server that is hosting the WebSphere Business Monitor server.

Archive

4. In the Preparing for the application installation panel (Figure A-6), ensure that **Show me all installation options and parameters** is selected and click **Next**.

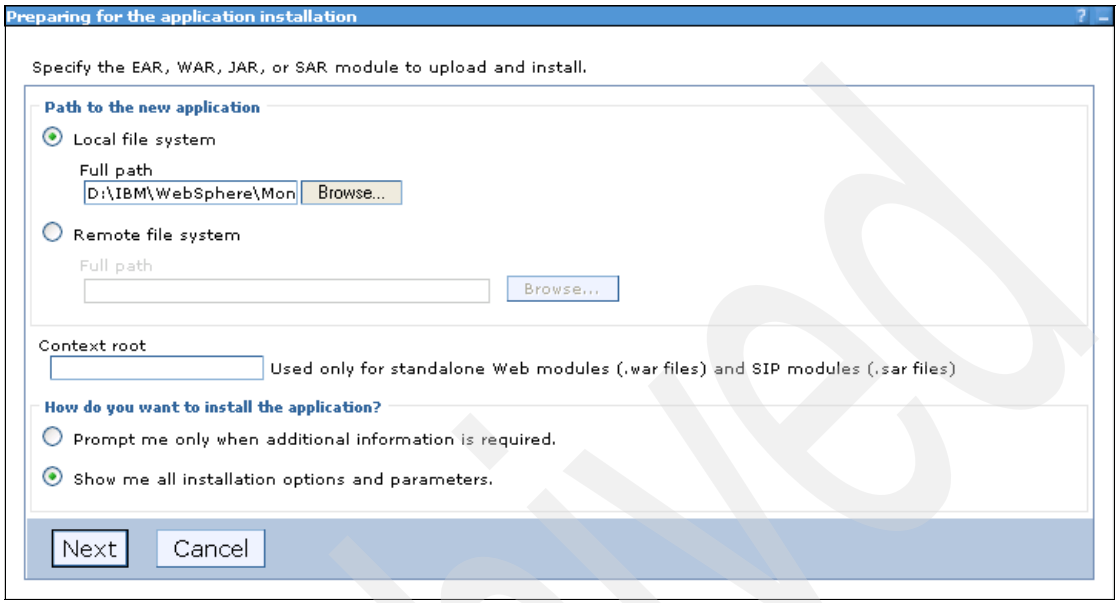


Figure A-6 Selecting GlobalHTMMApplication

5. Follow the instructions to install the monitor model in 3.4.1, “Installing a monitor model by using the default settings” on page 91.
6. Start the monitor model as explained in 3.4.3, “Starting and stopping a monitor model” on page 103.

After a successful installation, the GlobalHTMM is displayed as *Started* in the list of installed Monitor Models (Figure A-4 on page 742).

Connecting Business Process Choreographer with a WebSphere Business Monitor dashboard

To test the connection between the Web-based dashboard and WebSphere Process Server and to validate that WebSphere Process Server is running properly to use human task monitoring features:

1. Log in to the WebSphere Business Monitor administrative console.
2. In the administrative console (Figure A-7 on page 745), expand **Servers** and click **Application servers**. In the right pane, click **server1**.

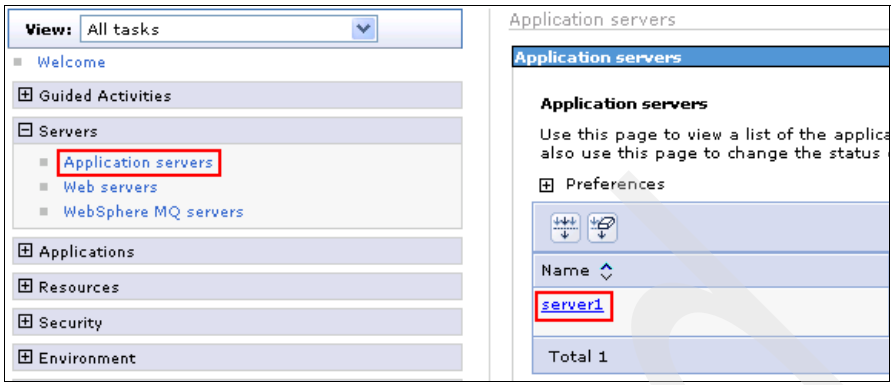


Figure A-7 Navigating to the WebSphere Business Monitor server

3. Under Server Infrastructure, expand **Java and Process Management** and click the **Process Definition** link (Figure A-8).

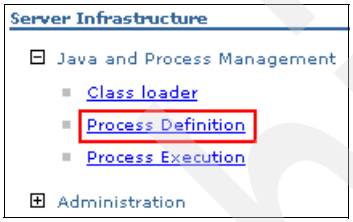


Figure A-8 Navigating to Process Definition

4. Under Additional Properties, click **Java Virtual Machine** (Figure A-9).

Figure A-9 Navigating to the Java Virtual Machine

5. Under Additional Properties, click **Custom Properties** (Figure A-10).

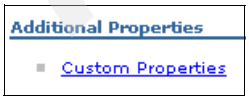


Figure A-10 Navigating to Custom Properties

Archive

6. The list of custom properties shows two variables (Figure A-11) called *DashboardBPCHost* and *DashboardBPCRMIPort*, which have no values defined. In case the two properties are not defined, create them by clicking the **New** button.

   a. Select the **DashboardBPCHost** property.

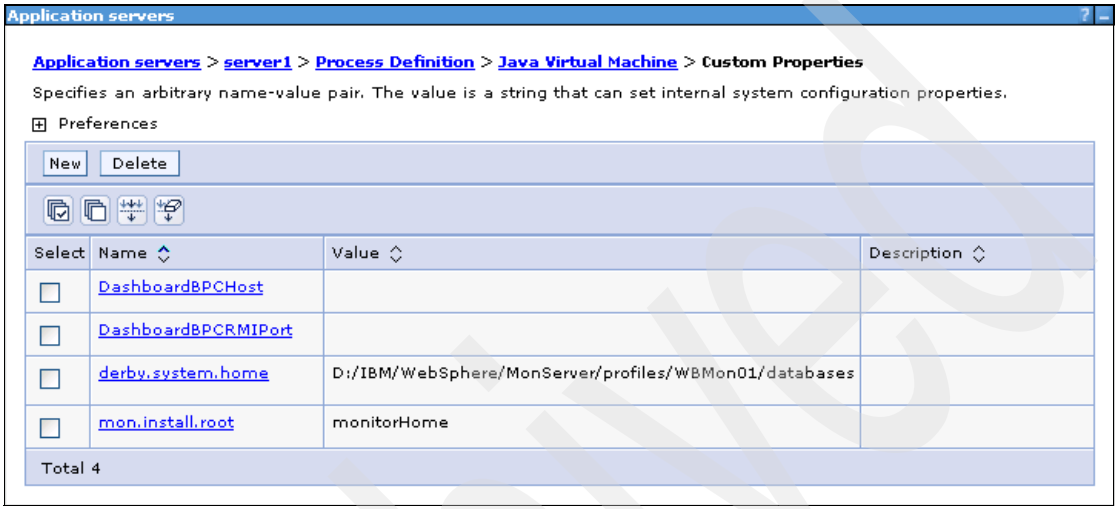


Figure A-11 WebSphere Business Monitor custom properties

   b. Under General Properties (Figure A-12), for Name, type the fully qualified host name of the WebSphere Process Server and click **OK**.

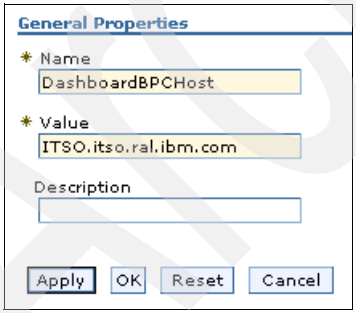


Figure A-12 Specifying the DashboardBPCHost host name

   c. Select the **DashboardBPCRMIPort** property.

Archived

d. Under General Properties (Figure A-13), enter the RMI port of the WebSphere Process Server and click **OK**.

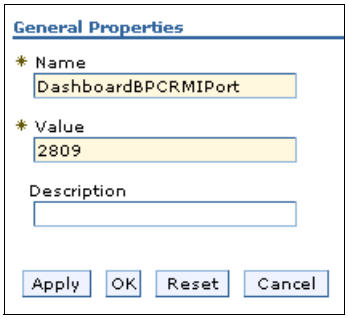


Figure A-13 Specifying the DashboardBPCRMIPort port

7. Save the settings to the master configuration.

Enabling security for WebSphere Business Monitor

To set up security for WebSphere Business Monitor:

1. Log in to the WebSphere Business Monitor administrative console.
2. Expand **Security** and click **Secure administration, applications and infrastructure**. In the right pane, click the **Security Configuration Wizard** button (Figure A-14).

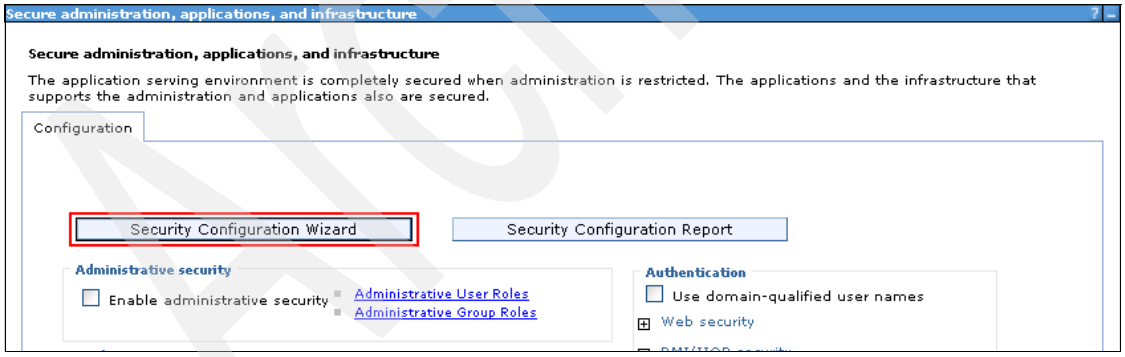


Figure A-14 Starting the Security Configuration Wizard for WebSphere Business Monitor

Archived

3. In the Step1: Specify extent of protection panel (Figure A-15), ensure that **Enable application security** is selected and click **Next**.

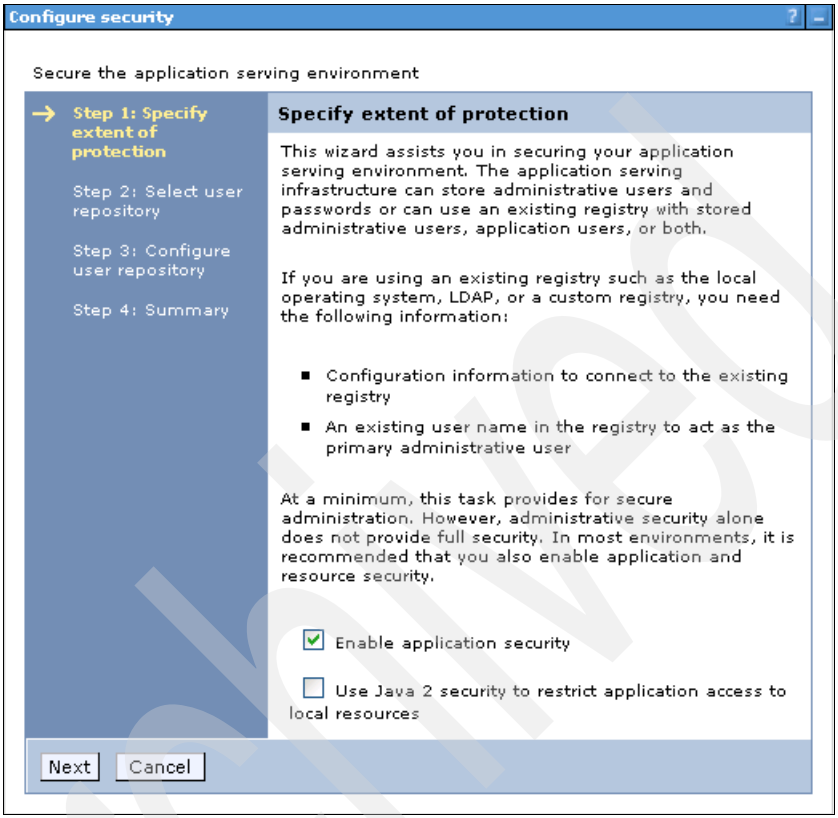


Figure A-15 Security Configuration Wizard: Step 1 for WebSphere Business Monitor

Archive

4. In the Step 2: Select user repository panel (Figure A-16), select **Federated repositories** and click **Next**.

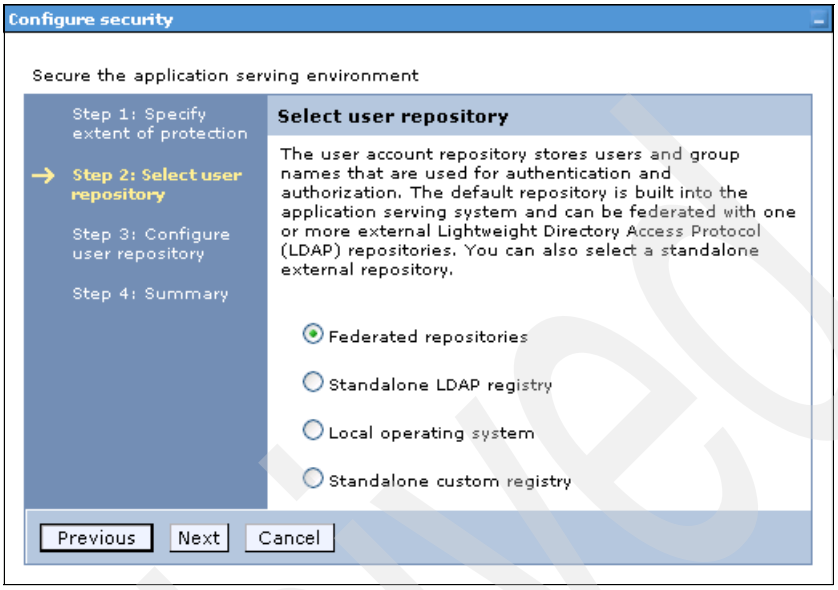


Figure A-16 Security Configuration Wizard: Step 2 for WebSphere Business Monitor

Archived

5. In the Step 3: Configure user repository panel (Figure A-17):
   a. For Primary administrative user name, type `wsadmin`.
   b. For Password, type `itso4you`.
   c. For Confirm password, type `itso4you`.
   d. Click **Next**.

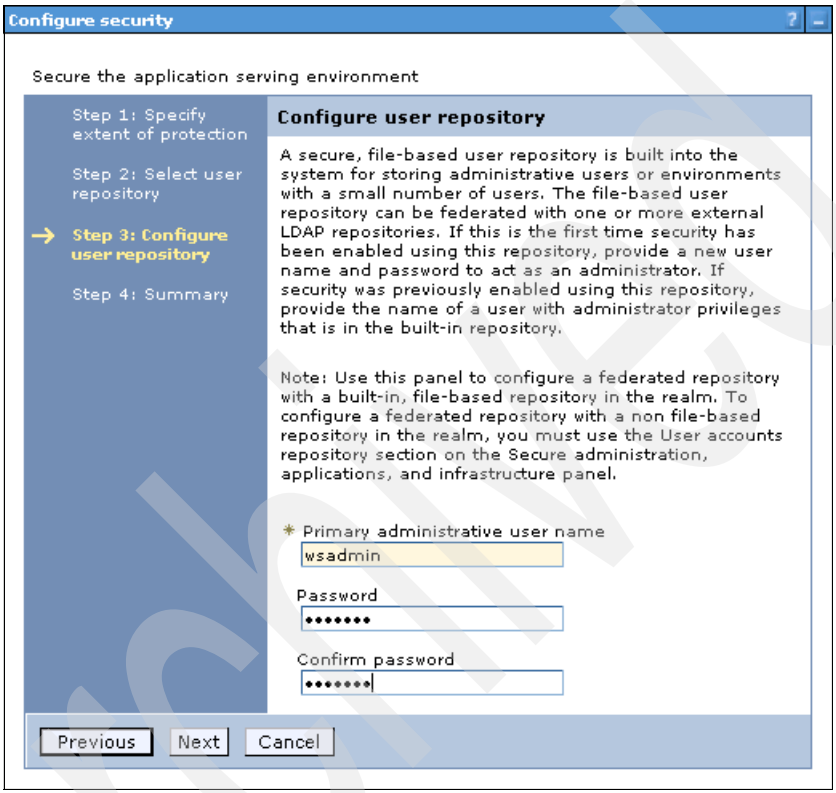


Figure A-17 Security Configuration Wizard: Step 3 for WebSphere Business Monitor

Archive

6. In the Step 4: Summary panel (Figure A-18), review the summary information and click **Finish**.

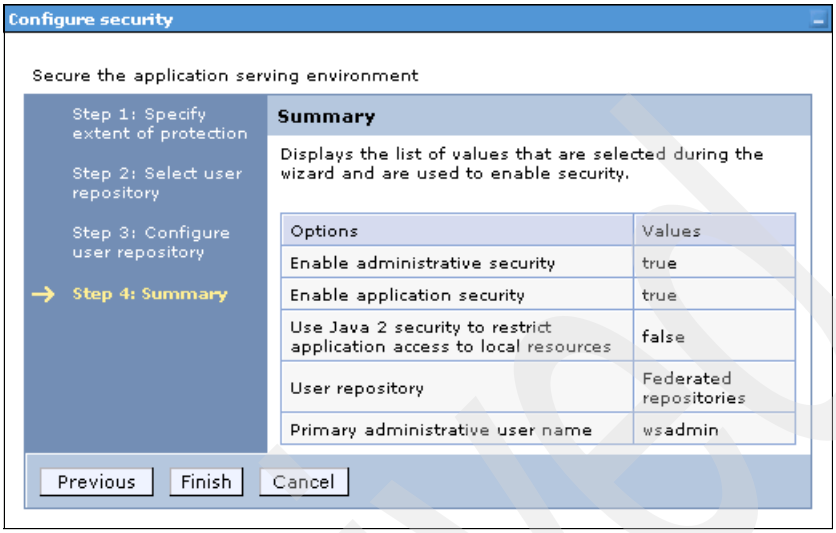


Figure A-18 Security Configuration Wizard: Step 4 for WebSphere Business Monitor

7. In the next panel (Figure A-19), for Available realm definitions, select **Federated repositories** and click the **Configure** button.

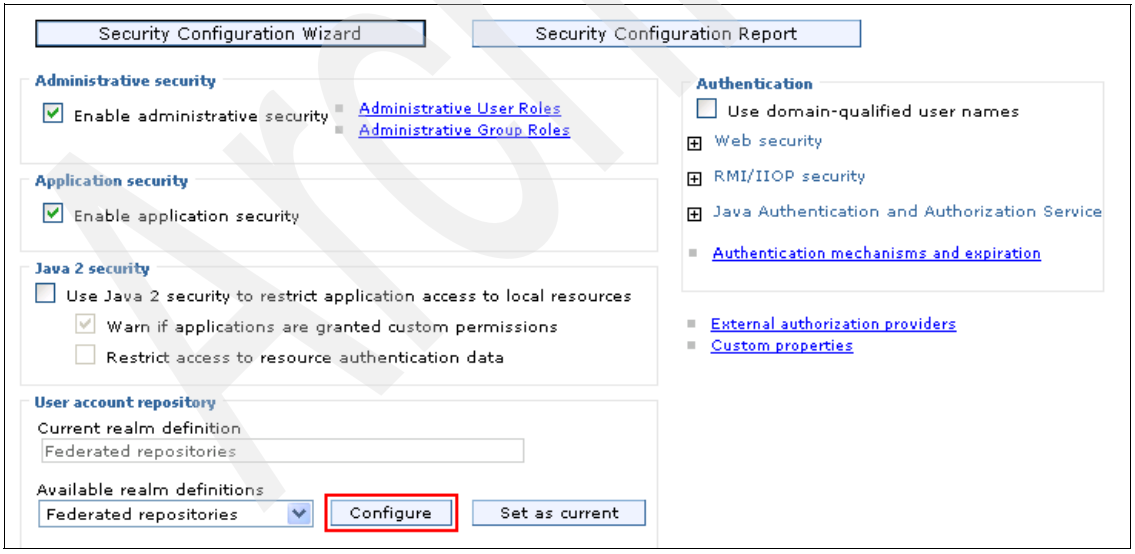


Figure A-19 Configuring the federated repositories for WebSphere Business Monitor

8. On the Configuration page, under General Properties (Figure A-20):
   a. For Realm name, type the name in the following format:

      `<fully qualified LDAP server host name>: <LDAP Port>`

      In this example, we type `9.42.170.182:389`.
   b. For Primary administrative user name, type `wsadmin`.
   c. Click the **Add Base entry to Realm** button.

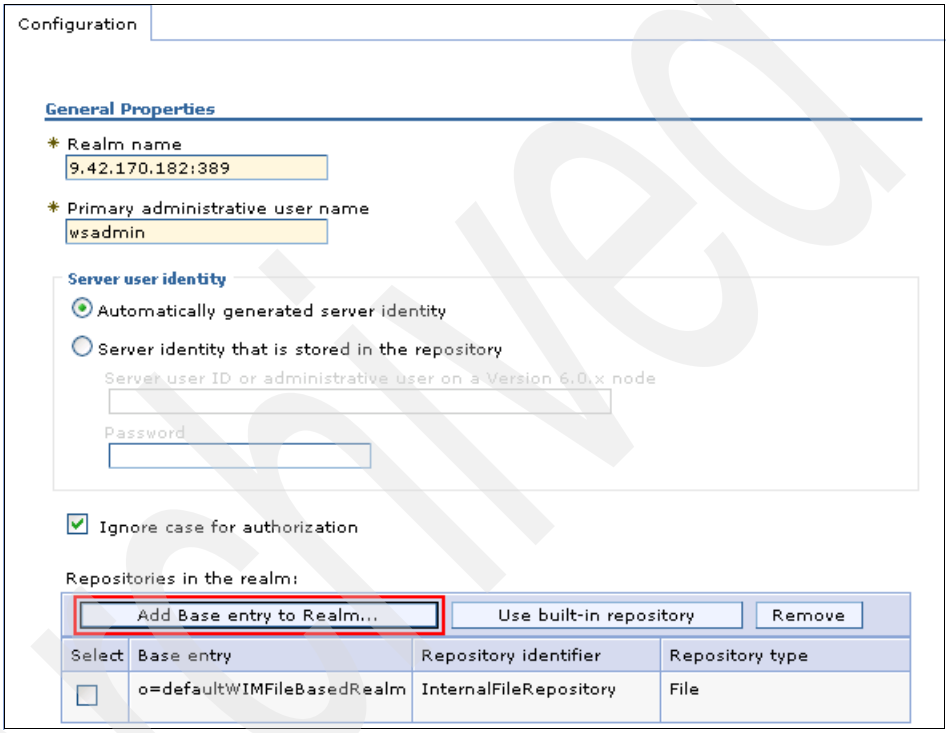


Figure A-20 Adding a base entry to realm for WebSphere Business Monitor

Archived

9. Click the **Add Repository** button (Figure A-21).

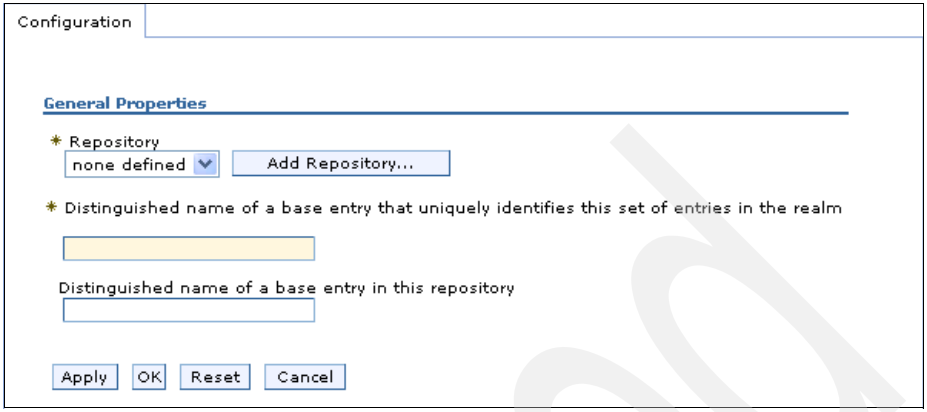


Figure A-21 Clicking the Add Repository button

10. On the Configuration page, under General Properties (Figure A-22 on page 754):
    a. For Repository identifier, type a meaningful identifier. In this example, we type `LDAP-Redbook`.
    b. For Directory type, select **IBM Tivoli Directory Server V6**.
    c. For Primary host name, type the host name of the LDAP server. In this example, we type `9.42.170.182`.
    d. For Port, type `389`.
    e. For Bind distinguished name, type `cn=root`.
    f. For Bind password, type `detlab0`.
    g. For Login properties, type `uid`.
    h. For Certificate mapping, select **EXACT_DN**.
    i. Click **OK**.

Archived

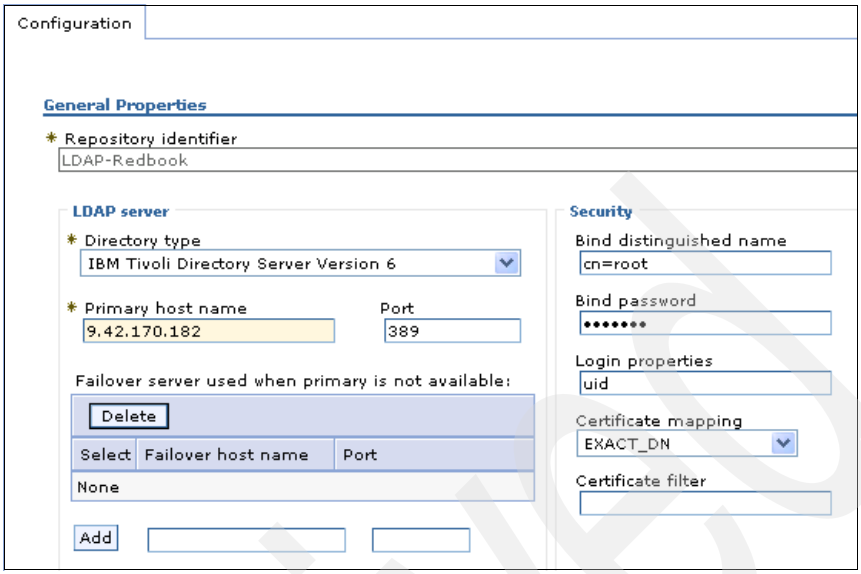


Figure A-22 Configuring the LDAP Repository for WebSphere Business Monitor

11. On the Configuration page, under General Properties (Figure A-23):
    a. For Repository, select **LDAP-Redbook**.
    b. For Distinguished names of a base entry that uniquely identifies this set of entries in the realm, type `o=customer`.
    c. For Distinguished name of a base entry in this repository, type `o=customer`.
    d. Click **OK**.

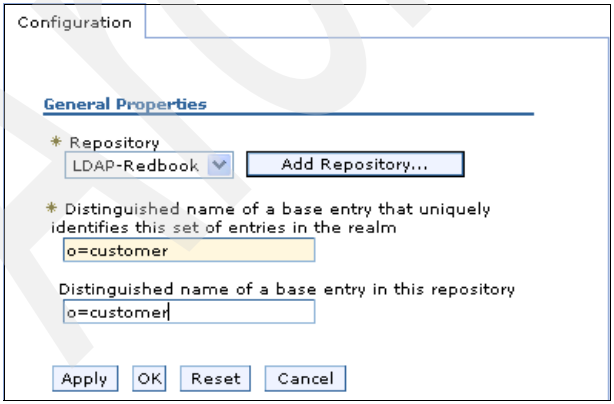


Figure A-23 Distinguished names for LDAP Repository for WebSphere Business Monitor

12. In the Federated repositories panel (Figure A-24), select **o=defaultWIMFileBasedRealm** (if it exists) and click the **Remove** button.

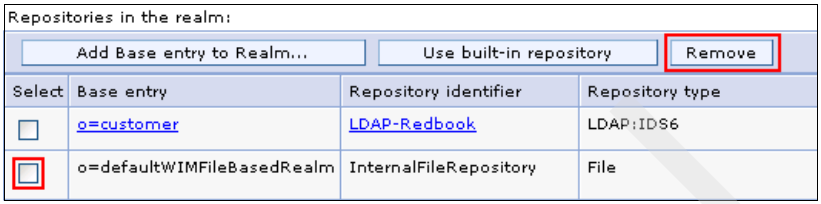


Figure A-24 Removing the default repository for WebSphere Business Monitor

13. Under Additional Properties, click **Supported entity types** (Figure A-25).

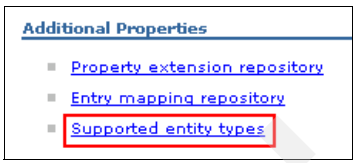


Figure A-25 Clicking the Supported entity types link

14. Update the Base entry for default parent values for Group, OrgContainer, and PersonAccount with `o=customer`. Accept the defaults for Relative Distinguished Name properties (Figure A-26).

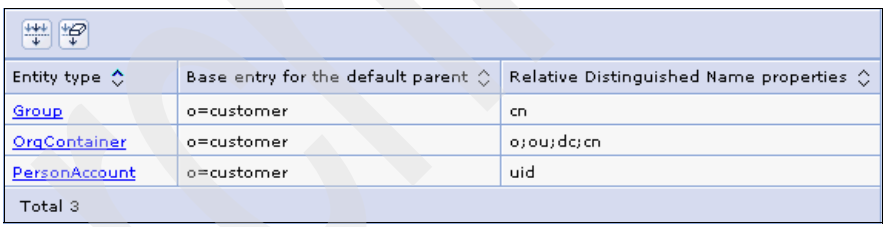


Figure A-26 Configuring the supported entity types for WebSphere Business Monitor

15. Save to the master configuration.

16. Navigate back to the Federated repositories panel (Figure A-27). Select **LDAP-Redbook**.

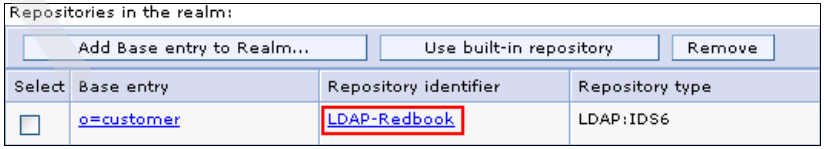


Figure A-27 Selecting the repository identifier for WebSphere Business Monitor

Archived

17. Under Additional Properties, click **LDAP entity types**.

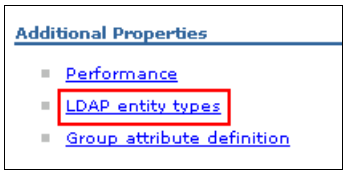


Figure A-28 Clicking LDAP entity types for WebSphere Business Monitor

18. Enter the following values as shown in Figure A-29:

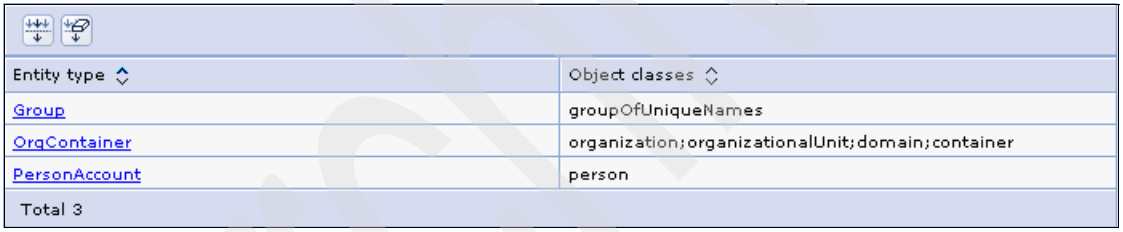
    a. For Group, specify the following values:

        i. For Object classes, type `groupOfUniqueNames`.
        ii. For Search bases, type `o=customer`.
        iii. For Search filter, type `(objectclass=groupOfUniqueNames)`.

    b. For OrgContainer, accept the default values.

    c. For PersonAccount, specify the following values:

        i. For Object classes, type `person`.
        ii. For Search bases, type `o=customer`.
        iii. For Search filter, type `(objectclass=person)`.



Entity type	Object classes
Group	groupOfUniqueNames
OrgContainer	organization;organizationalUnit;domain;container
PersonAccount	person

Total 3

Figure A-29 Configuring the LDAP entity types for WebSphere Business Monitor

19. Save to the master configuration.
20. Navigate back to the LDAP-Redbook Federated repository.
21. Under Additional Properties, click **Group attribute definition** (Figure A-30).

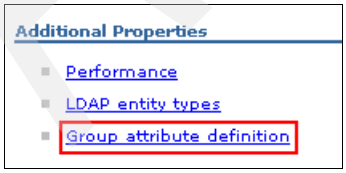


Figure A-30 Clicking Group attribute definition for WebSphere Business Monitor

Archived

22. On the Configuration page, under General Properties (Figure A-31):
    a. For Name of group membership attribute, type `LDAP-AllGroups`.
    b. Select **All - Contains all direct, nested and dynamic members**.
    c. Click **Apply**.

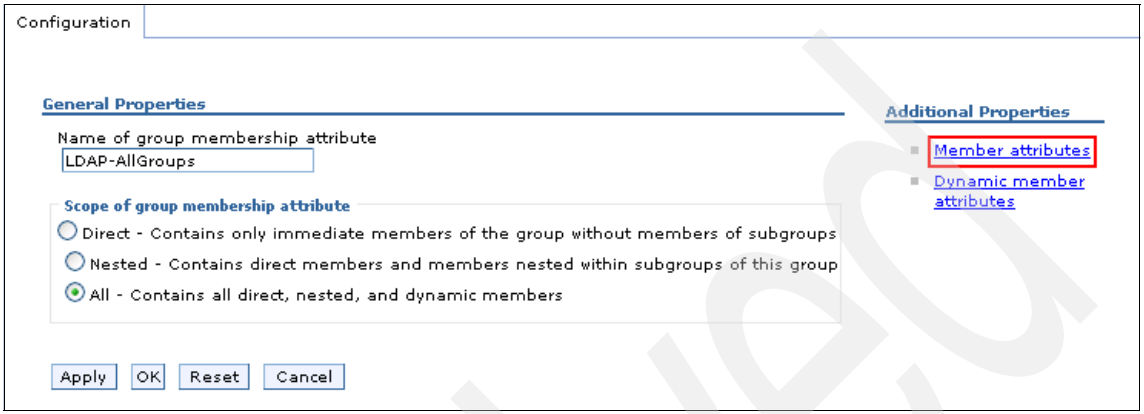


Figure A-31 Entering the group attribute definition for WebSphere Business Monitor

23. Under Additional properties and click **Member attributes**.
24. Click **New** to create a new member attribute (Figure A-32).

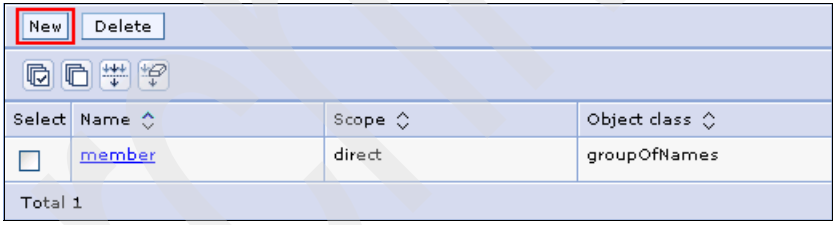


Figure A-32 Creating a new member for WebSphere Business Monitor

Archived

25. On the Configuration page, under General Properties (Figure A-33):
    a. For Name of member attribute, type `uniqueMember`.
    b. For Object class, type `groupOfUniqueNames`.
    c. Select **All - Contains all direct, nested and dynamic members**.
    d. Click **OK**.

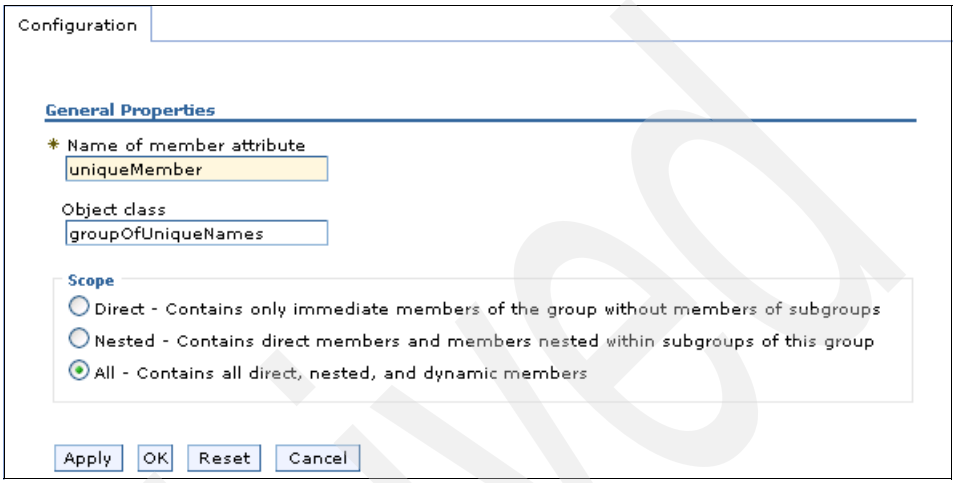


Figure A-33 Configuring a new member for WebSphere Business Monitor

26. Save to the master configuration.
27. Expand **Security** and select **Secure administration, applications and infrastructure**.
28. Under Authentication, click **Authentication mechanisms and expiration** (Figure A-34).

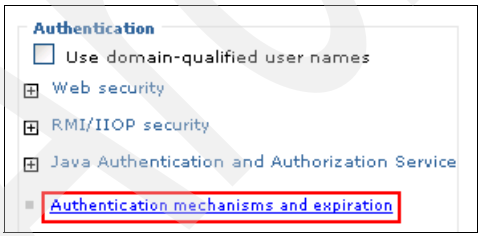


Figure A-34 Selecting Authentication mechanisms and expiration for WebSphere Business Monitor

29. On the Configuration page (Figure A-35):
    a. Click the **Generate Keys** button.
    b. Under Cross-cell single sign-on:

Archived

i. For Password, type `password`.

ii. For Confirm password, type `password`.

iii. For Fully Qualified Key file name, type the location of where to export the key. In this example, we type `D:\secureMonitor.key`.

iv. Click the **Export Keys** button.

c. Click **OK**.

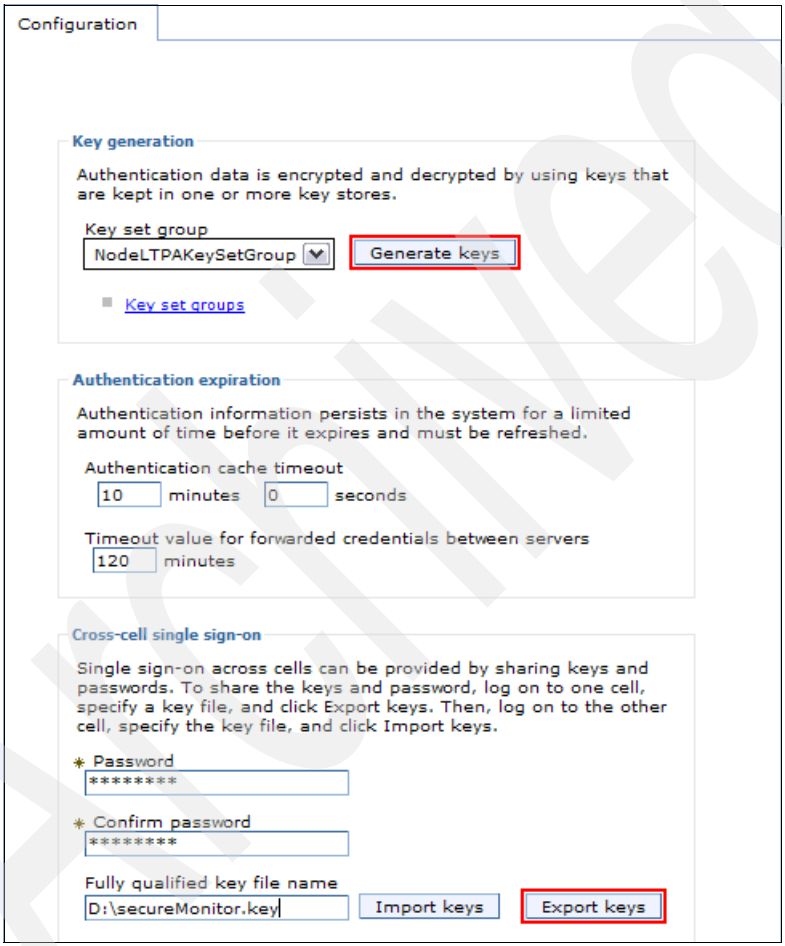


Figure A-35 Exporting a security key file

30. Save to the master configuration.

31. Restart the WebSphere Business Monitor server.

Archived

Enabling security for WebSphere Business Monitor applications

To enable security for WebSphere Business Monitor applications:

1. Launch the WebSphere Business Monitor administrative console. Log in with a user name of `wsadmin` and a password of `itso4you`.
2. Expand **Applications** and click **Enterprise Applications**.
3. In the Enterprise Application panel (Figure A-36), click the **AlphabloxPlatform** application.

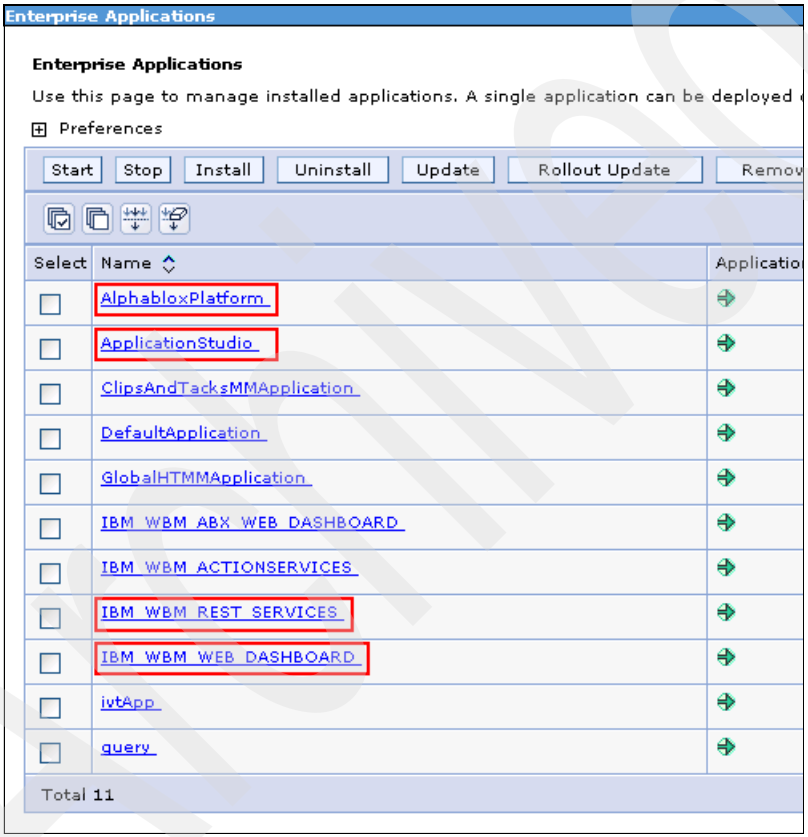


Figure A-36 Selecting WebSphere Business Monitor applications

Archived

4. Under Detail Properties, click **Security role to user/group mapping** (Figure A-37).

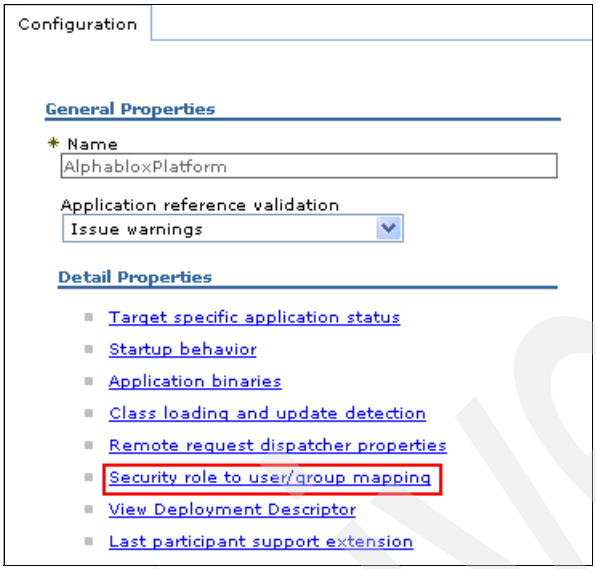


Figure A-37 Navigating to Security role to user/group mapping for WebSphere Business Monitor

5. Select the **AlphabloxAdministrator** check box and click the **Look up users** button (Figure A-38).

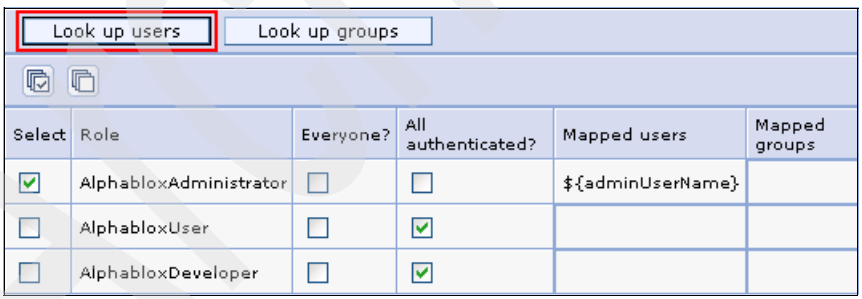


Figure A-38 Looking up user for AlphabloxAdministrator

Archived

6. In the next panel (Figure A-39):
   a. For Search String, type an asterisk (*) for wildcard and click the **Search** button.
   b. Under Available, select **wsadmin**, and click the **>>** button to move the user to the selected text area.
   c. If it exists, under Selected, select **${adminUserName}** and click the **<<** button to remove the user.
   d. Click **OK**.

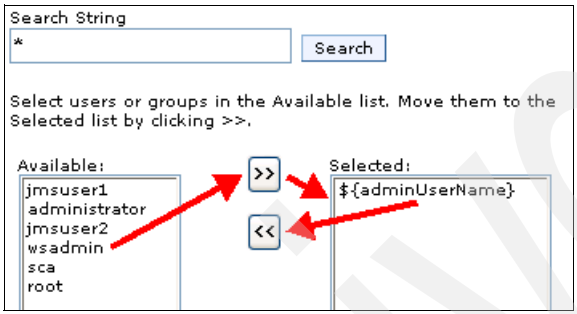


Figure A-39 Selecting a user for Alphablox Administrator

7. For AlphabloxUser and AlphabloxDeveloper, select the **All authenticated?** check boxes (Figure A-38 on page 761). Click **OK**.
8. Save to the master configuration.
9. In the administrative console, expand **Applications** and click **Enterprise Applications**. In the right pane, click the **ApplicationStudio** application (Figure A-36 on page 760).
10. Under Detail Properties, click **Security role to user/group mapping**.
11. Map the **wsadmin** user for AlphabloxAdministrator and delete the **${adminUserName}** user (Figure A-39).
12. For AlphabloxUser, select the **All authenticated?** check box (Figure A-40).

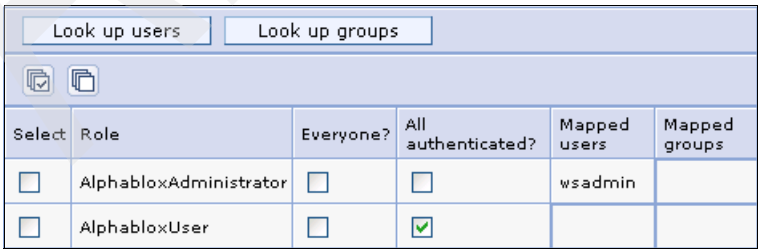


Figure A-40 Selecting a user for the ApplicationStudio application

Archived

13. Click **OK**.
14. Save to the master configuration.
15. In the administrative console, expand **Applications** and click **Enterprise Applications**. In the right pane, click the **IBM_WBM_REST_SERVICES** application (Figure A-36 on page 760).
16. Under Detail Properties, click **Security role to user/group mapping**.
17. For monitorusers, select the **All authenticated?** check box (Figure A-41).

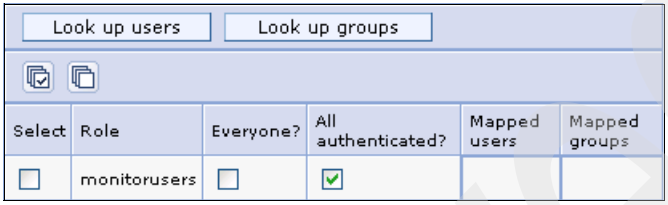


Figure A-41 Selecting a user for the IBM_WBM_REST_SERVICES application

18. Click **OK**.
19. Save the changes to the master configuration.
20. In the administrative console, expand **Applications** and click **Enterprise Applications**. In the right pane, click **ApplicationStudio** and **IBM_WBM_WEB_DASHBOARD** (Figure A-36 on page 760).
21. Under Detail Properties, click **Security role to user/group mapping**.
22. For Administrator, select the **All authenticated?** check box (Figure A-42).

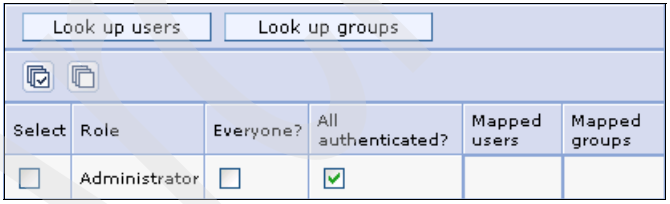


Figure A-42 Selecting a user for IBM_WBM_WEB_DASHBOARD application

23. Click **OK**.
24. Save to the master configuration.

Archived

Updating the Alphablox Server.properties file

Update the server.properties file of Alphablox with the credentials of the wsadmin user:

1. Stop WebSphere Business Monitor. Because security is enabled, enter the following statement on a command line from the bin directory of the WebSphere Business Monitor node:

   ```
   stopServer.bat server1 -username wsadmin -password itso4you
   ```

   Attention: Ensure that the WebSphere Business Monitor server is stopped for the following actions. If it is started, the changes part of the next steps will be overwritten when WebSphere Business Monitor server is stopped.

2. Navigate to the <WBM_PROFILE>\Alphablox_server1\repository\servers\AlphabloxAnalytics folder.
3. Edit the server.properties file by adding the following values at the end of the file:

   ```
   ws.admin.username = wsadmin
   ws.admin.password = itso4you
   ```

4. Start the WebSphere Business Monitor server by entering the following command in the profiles bin directory:

   ```
   startServer.bat server1
   ```

5. Ensure that the WebSphere Business Monitor server is started correctly and review the SystemOut.log file for security related error messages.

   Note: The SystemOut.log file of the WebSphere Business Monitor server is in <WBM_PROFILE>\logs.

Updating the J2C authentication data entries for messaging buses for WebSphere Business Monitor

To set up security for the messaging buses of WebSphere Business Monitor:

1. Launch the WebSphere Business Monitor administrative console and log in with a user name of `wsadmin` and a password of `itso4you`.
2. Expand **Security** and click **Bus Security**.

Archive

3. For the CommonEventInfrastructure_Bus and MONITOR.monitor03Node01Cell.Bus buses:

   a. Click the **Disabled** link of the current bus (Figure A-43).

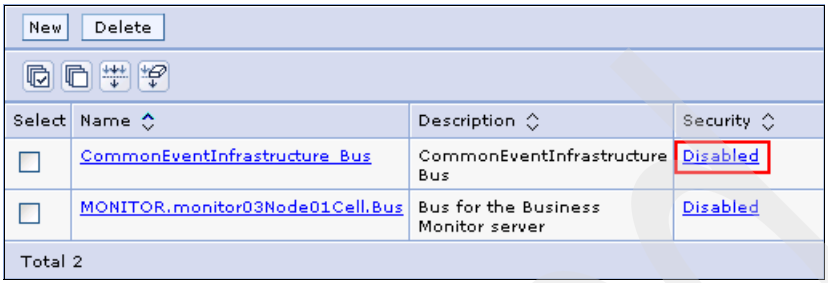


Figure A-43 Selecting the Disabled link of a WebSphere Business Monitor bus

   b. Under Relation Items, click **JAAS-J2C authentication data** (Figure A-44).

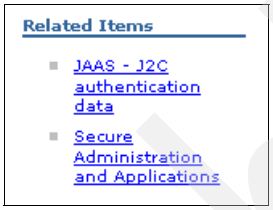


Figure A-44 Selecting JAAS-J2C authentication data for the WebSphere Business Monitor bus

Archive

c. Click the **MonitorAlphabloxAlias** alias (Figure A-45).

Figure A-45 WebSphere Business Monitor Aliases

Alias names: The aliases may have different names in another environment.

d. Enter the following values for the alias (Figure A-46):

i. For User ID, type `wsadmin`.
ii. For Password, type `itso4you`.

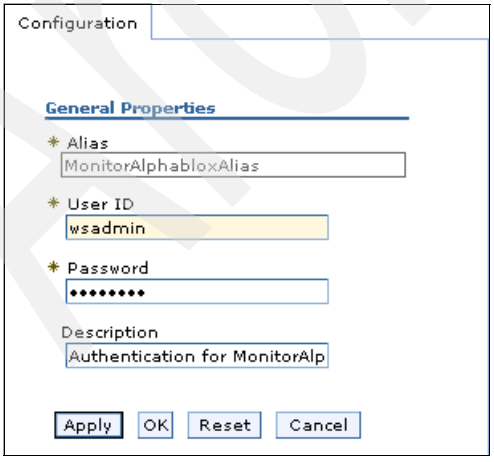


Figure A-46 Configuring aliases for the WebSphere Business Monitor bus

e. Repeat steps c and d on page 766 for each of the following aliases:

- MonitorBusAuth
- MonitorBusLinkAuth
- MonitorQueueConnectionFactoryAuth

f. Save to the master configuration.

g. Navigate back to the **Disabled** link of the current bus. Under Additional Properties, click **Users and groups in the bus connector role** (Figure A-47).

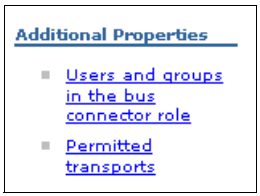


Figure A-47 Navigating to Users and groups in the bus connector role for WebSphere Business Monitor

h. Click the **New** button to add a new user (Figure A-48).

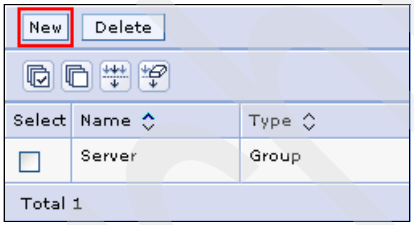


Figure A-48 Creating a new bus connector role for WebSphere Business Monitor

i. On the Configuration page under General Properties (Figure A-49), select **User name** and type `wsadmin`. Click **OK**.

Figure A-49 Configuring the bus connector role for WebSphere Business Monitor

j. Save to the master configuration.

k. Navigate back to the **Disabled** link of the current bus.

l. On the Configuration page, under General Properties (Figure A-50):

   i. Select the **Enable bus security**.
   ii. Inter-engine authentication alias, select **MonitorBusAlias**.
   iii. Click **OK**.

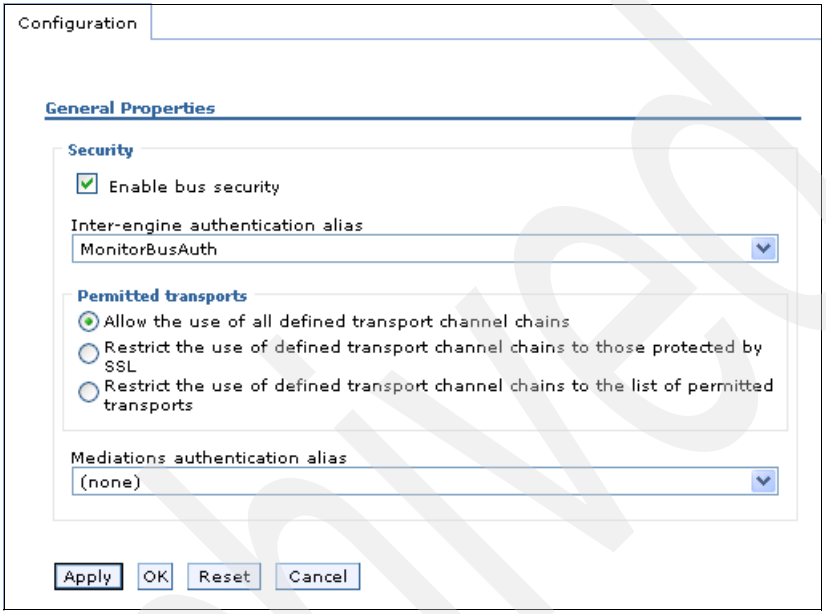


Figure A-50 Enabling WebSphere Business Monitor bus security

m. Save to the master configuration.

Archived

Enabling security for WebSphere Business Monitor resources

To enable security for WebSphere Business Monitor resources:

1. In the WebSphere Business Monitor administration console, expand **Resources** → **JMS** and click **Topic connection factories** (Figure A-51). In the right pane, select **CommonEventInfrastructure_AllEventsTopicCF**.

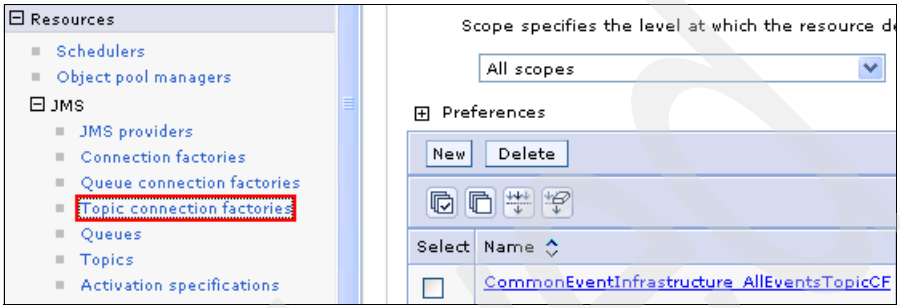


Figure A-51 Navigating to Topic connection factories for WebSphere Business Monitor

2. Scroll down. For Component authentication alias, select **MonitorBusAuth** (Figure A-52). Click **OK**.

Figure A-52 Entering an alias for Topic connection factory for WebSphere Business Monitor

Archive

3. In the WebSphere Business Monitor administration console, expand **Resources** → **JMS** and click **Activation specification** (Figure A-53). In the right pane, select **CommonEventInfrastructure_ActivationSpec**.

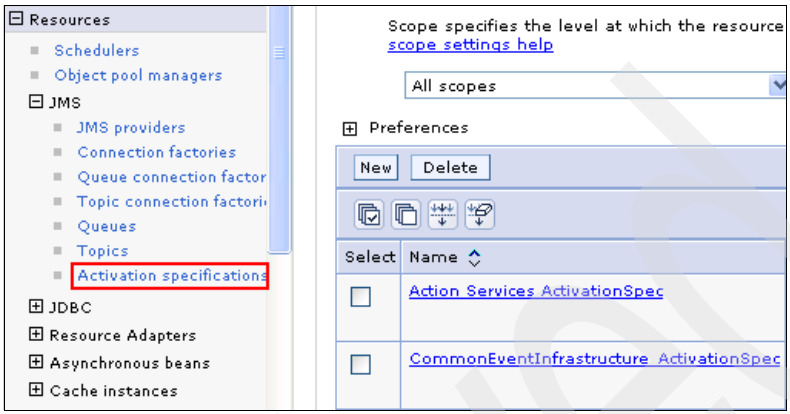


Figure A-53 Navigating to Activation specifications for WebSphere Business Monitor

4. Scroll down. For Authentication alias, select **MonitorBusAuth**. Click **OK**.
5. Save to the master configuration.

Modifying the Alert User IDs

Adapt all created alerts to the new set of users provided by the LDAP server:

1. Log in to the WebSphere Business Monitor administrative console.
2. Expand **Applications** → **Monitor Action Services** → **Template Definitions** and click **Notifications**.
3. For each notification (Figure A-54):
   a. Click the notification.
   b. In the 'To' query type field, select **User id**, and for To, type `wsadmin`.

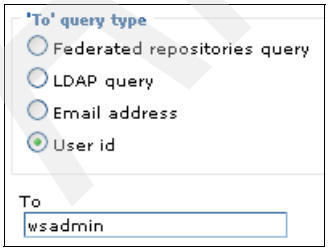


Figure A-54 Modifying the notification properties

4. Click **OK**.

Enabling data security for monitor models

If security is enabled for a WebSphere Business Monitor server the monitor model has to be secured as well. Monitor models can be grouped into resource groups to handle data access permissions. Permissions must be assigned to a resource group. A resource group has exactly one role, of the following four available roles, and consists of a set of users:

- Business-Manager

  This role provides basic access.

- Personal-KPI-Administrator

  This role gives users the authority to create nonshared (personal) key performance indicators (KPIs). Only the owner and KPI-administrator can view and update the created KPIs. Users of this role have read access to all metadata associated with the model except KPIs.

- Public-KPI-Administrator

  This role gives users the authority to create shared (public) or nonshared (personal) KPIs. Other users can use and view shared (public) KPIs. Only the owner or KPI-Administrator can make changes to shared (public) KPIs. Users of this role have read access to all metadata associated with the model, except the KPIs.

- KPI-Administrator

  This role gives users all the authority associated with KPI administration. Users of this role can create shared (public) and nonshared (personal) KPIs. In addition, a KPI-Administrator can change the ownership of any KPIs. This role can view all KPIs, shared and nonshared. Users of this role can update noncore properties and have read access to all metadata associated with the model, except the KPIs.

The root resource group is the default group. The wsadmin user is added to the BusinessManager and KPI-Administrator role of all monitor models:

1. In the WebSphere Business Monitor administration console, expand **Security** and click **Monitor Data Security**. In the right pane, click the **root** link (Figure A-55 on page 772).

Archived

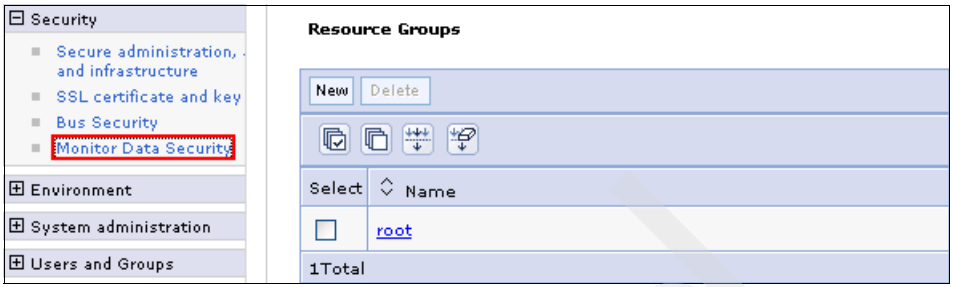


Figure A-55 Navigating to Monitor Data Security

2. Select **Business-Manager** and click the **Users** button (Figure A-56).

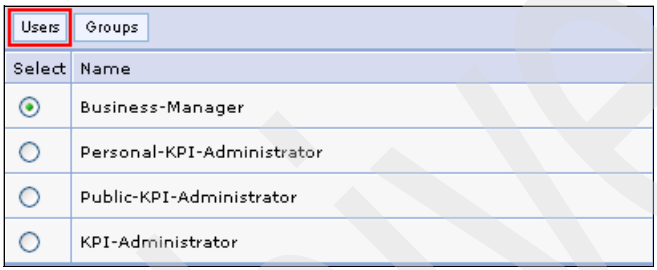


Figure A-56 Adding a user to the Business-Manager role

3. In the next panel (Figure A-57):
   a. Click the **Search** button.
   b. Under Available, select the **wsadmin** user and click the **>** button to move the user to the Selected box.

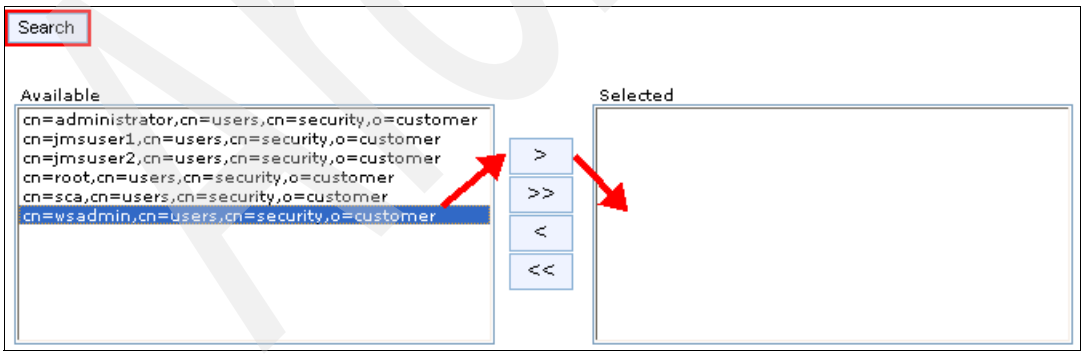


Figure A-57 Configure Business-Manager user

   c. Click **OK**.

4. Select **KPI-Administrator** and click the **Users** button (Figure A-58).

Figure A-58 Add user to KPI-Administrator role

5. In the next panel:
   a. Click the **Search** button.
   b. Under Available, select the **wsadmin** user and click the > button to move the user to the Selected box.
   c. Click **OK**.
6. Restart the WebSphere Business Monitor server by executing the following commands on a command line from the bin directory of the WebSphere Business Monitor node:

```
stopServer.bat server1 -username wsadmin -password itso4you
startServer.bat server1
```

Enabling security for WebSphere Process Server

In this section, we set up security for WebSphere Process Server. Later we export the key file from WebSphere Business Monitor and copy it to the server that is hosting WebSphere Process Server.

1. Log in to the WebSphere Process Server administrative console.
2. Expand **Security** and click **Secure administration, applications and infrastructure**. In the right pane, click the **Security Configuration Wizard** button (Figure A-59).

Figure A-59 Starting the Security Configuration Wizard for WebSphere Process Server

Archived

3. In Step 1, select **Enable application security** and click **Next** (Figure A-60).

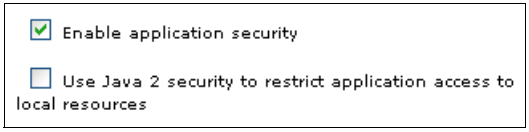


Figure A-60 Security Configuration Wizard: Step 1 for WebSphere Process Server

4. In Step 2, select **Federated repositories** and click **Next** (Figure A-61).

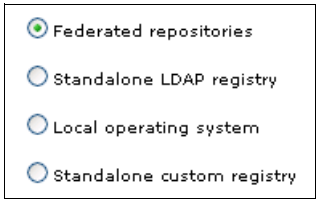


Figure A-61 Security Configuration Wizard: Step 2 for WebSphere Process Server

5. In Step 3, enter the following values (Figure A-62):
   a. For Primary administrative user name, type `wsadmin`.
   b. For Password, type `itso4you`.
   c. For Confirm password, type `itso4you`.
   d. Click **Next**.

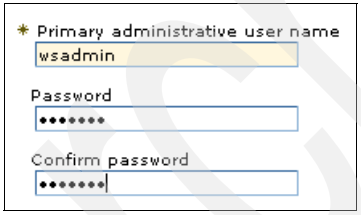


Figure A-62 Security Configuration Wizard: Step 3 for WebSphere Process Server

6. In Step 4, review the summary information and click **Finish** (Figure A-63).

Options	Values
Enable administrative security	true
Enable application security	true
Use Java 2 security to restrict application access to local resources	false
User repository	Federated repositories
Primary administrative user name	wsadmin

Figure A-63 Security Configuration Wizard: Step 4 for WebSphere Process Server

Archived

7. For Available realm definitions, select **Federated repositories** and click the **Configure** button (Figure A-64).

Figure A-64 Configuring the Federated Repositories for WebSphere Process Server

8. On the Configuration page, under General Properties:
   a. For Realm name, type the name in the following format:

      `<fully qualified LDAP server host name>: <LDAP Port>`

      In this example, we type `9.42.170.182:389`.

   b. For Primary administrative user name, type `wsadmin`.

   c. Click the **Add Base entry to Realm** button (Figure A-65).

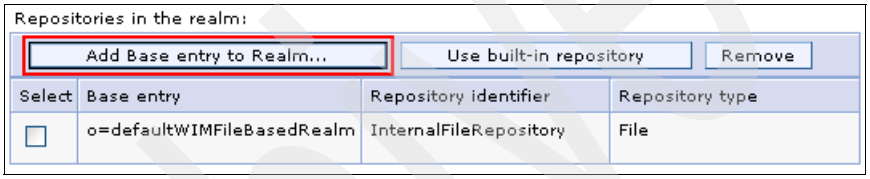


Figure A-65 Adding a base entry to realm for WebSphere Process Server

9. Click the **Add Repository** button (Figure A-66).

Figure A-66 Adding a repository for WebSphere Process Server

Archived

10. On the Configuration page, under General Properties (Figure A-67):
    a. For Repository identifier, type a meaningful identifier. We type `LDAP-Redbook`.
    b. For Directory Type, select **IBM Tivoli Directory Server V6**.
    c. For Primary host name, type the host name of the LDAP server. We type `9.42.170.182`.
    d. For Port, type `389`.
    e. For Bind distinguished name, type `cn=root`.
    f. For Bind password, type `detlab0`.
    g. For Login properties, type `uid`.
    h. For Certificate mapping, select **EXACT_DN**.

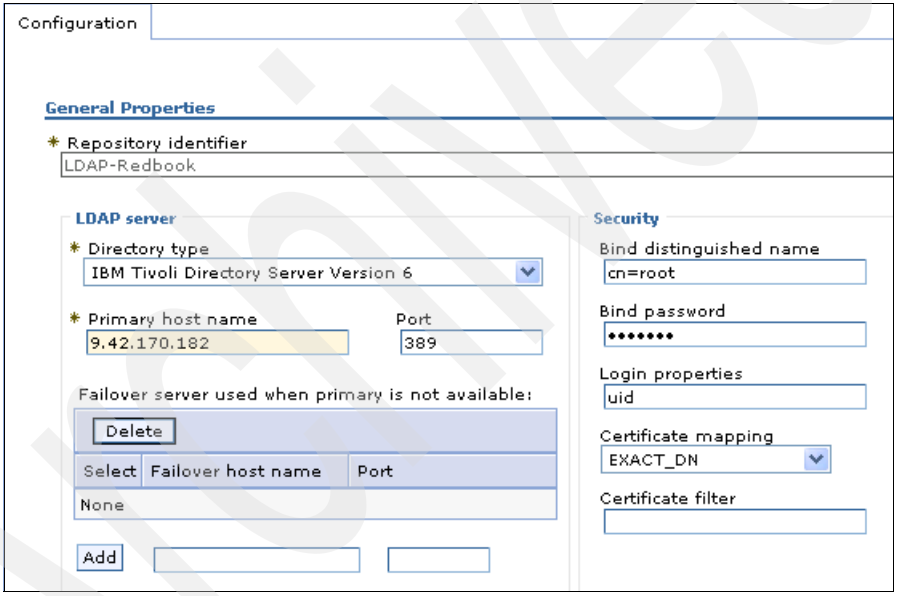


Figure A-67 Configuring the LDAP Repository for WebSphere Process Server

    i. Click **OK**.

Archived

11. On the Configuration page, under General Properties (Figure A-68):
    a. For Repository, select **LDAP-Redbook**.
    b. For Distinguished names of a base entry that uniquely identifies this set of entries in the realm, type `o=customer`.
    c. For Distinguished name of a base entry in this repository, type `o=customer`.

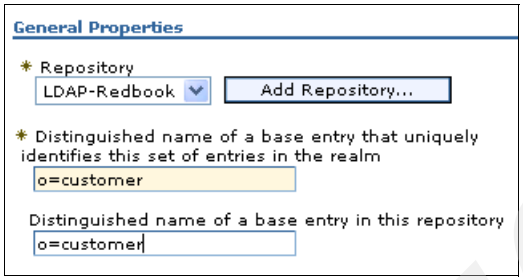


Figure A-68 Entering Distinguished names for LDAP Repository for WebSphere Process Server

    d. Click **OK**.

12. Back in the Federated repositories panel (Figure A-24), select **o=defaultWIMFileBasedRealm** (if it exists) and click the **Remove** button.

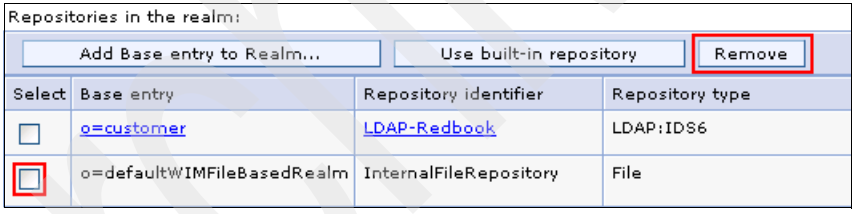


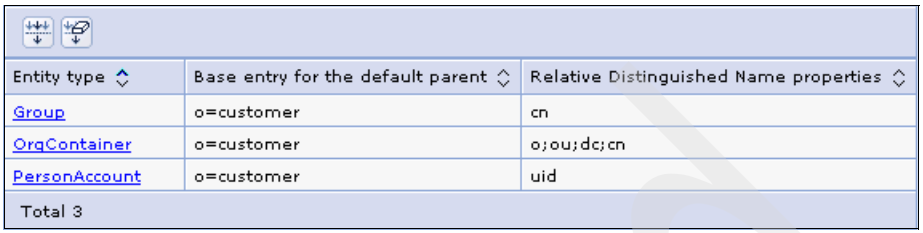
Figure A-69 Remove default repository for WebSphere Process Server

13. Under Additional Properties, click **Supported entity types** (Figure A-70).

Figure A-70 Selecting Supported entity types for WebSphere Process Server

Archived

14. Update the Base entry for default parent values for Group, OrgContainer, and PersonAccount with `o=customer`. Accept the defaults for Relative Distinguished Name properties (Figure A-71).



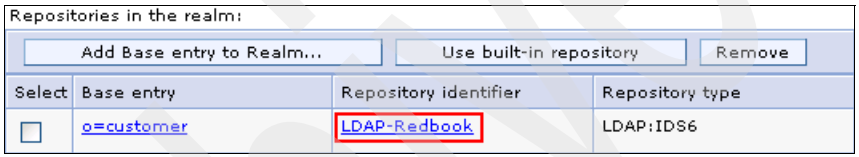
Entity type	Base entry for the default parent	Relative Distinguished Name properties
Group	o=customer	cn
OrgContainer	o=customer	o;ou;dc;cn
PersonAccount	o=customer	uid

Total 3

Figure A-71 Configuring the supported entity types for WebSphere Process Server

15. Save to the master configuration.

16. Navigate to the Federated repositories panel (Figure A-72) and click **LDAP-Redbook**.



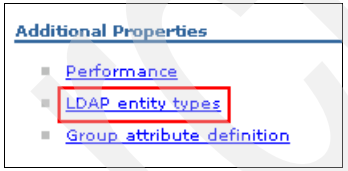
Repositories in the realm:

Add Base entry to Realm... | Use built-in repository | Remove

Select	Base entry	Repository identifier	Repository type
☐	o=customer	LDAP-Redbook	LDAP:IDS6

Figure A-72 Selecting the Repository identifier for WebSphere Process Server

17. Under Additional Properties, click **LDAP entity types**.



Additional Properties

- Performance
- LDAP entity types
- Group attribute definition

Figure A-73 Selecting LDAP entity types for WebSphere Process Server

Archive

18. Enter the following values as shown in Figure A-74:
    a. For Group, specify the following values:
        i. For Object classes, type `groupOfUniqueNames`.
        ii. For Search bases, type `o=customer`.
        iii. For Search filter, type `(objectclass=groupOfUniqueNames)`.
    b. For OrgContainer, accept the default values.
    c. For PersonAccount, specify the following values:
        i. For Object classes, type `person`.
        ii. For Search bases, type `o=customer`.
        iii. For Search filter, type `(objectclass=person)`.

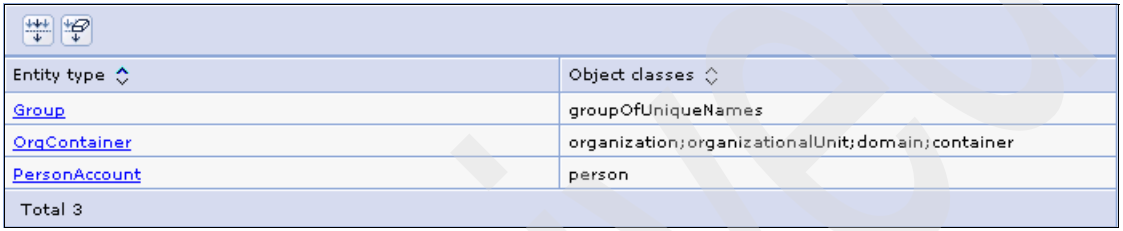


Figure A-74 Configure LDAP Entity Types for WebSphere Process Server

19. Save to the master configuration.
20. Navigate back to the **LDAP-Redbook** Federated repositories.
21. Under Additional Properties, select **Group attribute definition** (Figure A-75).

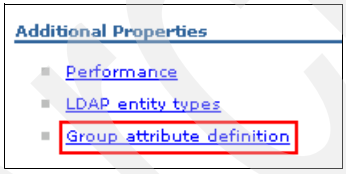


Figure A-75 Selecting Group attribute definition for WebSphere Process Server

22. Under General Properties (Figure A-76 on page 780):
    a. For Name of group membership attribute, type `LDAP-AllGroups`.
    b. Select **All - Contains all direct, nested and dynamic members**.
    c. Click **Apply**.
    d. Under Additional properties, click **Member attributes**.

Archived

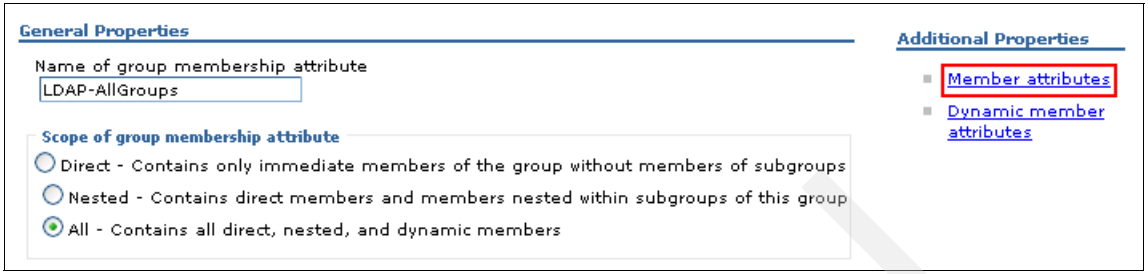


Figure A-76 Entering the group attribute definition for WebSphere Process Server

23. Click **New** to create a new member attribute (Figure A-77).

Figure A-77 Creating a new member for WebSphere Process Server

24. Under General Properties (Figure A-78):

    a. For Name of member attribute, type `uniqueMember`.
    b. For Object class, type `groupOfUniqueNames`.
    c. Select **All - Contains all direct, nested and dynamic members**.

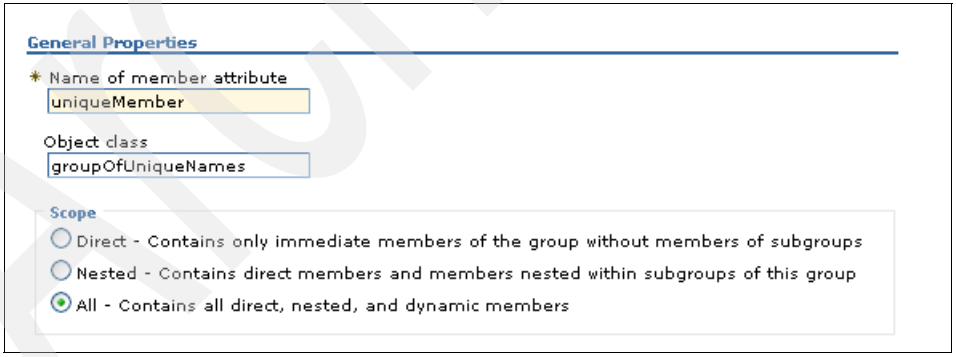


Figure A-78 Configuring a new member for WebSphere Process Server

    d. Click **OK**.

25. Save to the master configuration.

26. In the administrative console, expand **Security** and click **Secure administration, applications and infrastructure**.

27. Under Authentication, click **Authentication mechanisms and expiration** (Figure A-79).

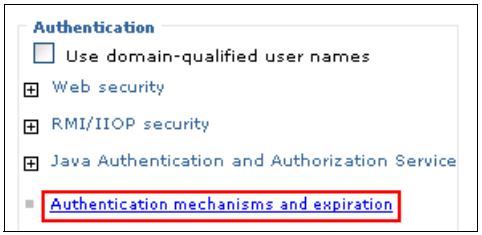


Figure A-79 Selecting Authentication mechanisms and expiration for WebSphere Process Server

28. Under Cross-cell single sign-on (Figure A-80):
    a. For Password, type `password`.
    b. For Confirm password, type `password`.

> **Note:** The password must match the one that you entered for WebSphere Business Monitor when exporting the file.

    c. For Fully Qualified Key file name, type the path where you stored the key file. In this example, we type `D:\secureMonitor.key`.
    d. Click the **Import Keys** button.

Figure A-80 Importing a security key file

    e. Click **OK**.

29. Save to the master configuration.
30. Restart WebSphere Process Server:
    a. In WebSphere Integration Developer, click the **Servers** tab.
    b. Double-click **WebSphere Process Server v6.1** to open the Properties view.

Archived

c. Navigate to Security (Figure A-81):

i. Ensure that **Security is enabled on this server** is selected.
ii. Change User ID to `wsadmin` and Password to `itso4you`.

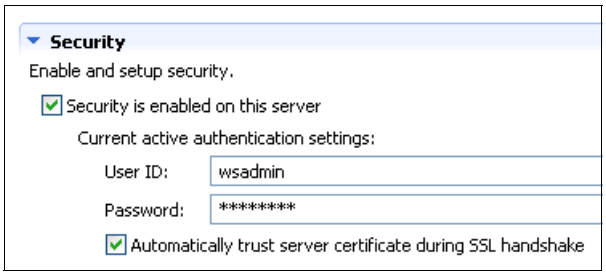


Figure A-81 Changing the WebSphere Process Server login properties

d. Press Ctrl+S to save your work.

e. Restart the server by using WebSphere Integration Developer or by executing the following commands from the WebSphere Process Server profiles bin directory:

```
stopServer.bat server1 -username wsadmin -password itso4you
startServer.bat server1
```

Note: Error messages are displayed in the SytsemOut.log because the security setup is not completed yet.

Enabling security for WebSphere Process Server applications

To enable security for WebSphere Process Server applications:

1. Launch the WebSphere Process Server administrative console and log in with `wsadmin` as the user name and `itso4you` as the password.
2. Expand **Applications** and click **Enterprise Applications**.

Note: Not all application are started because the security setup is not completed yet.

Archived

3. Click the **BPEContainer_<cellname>_server1** application (Figure A-82).

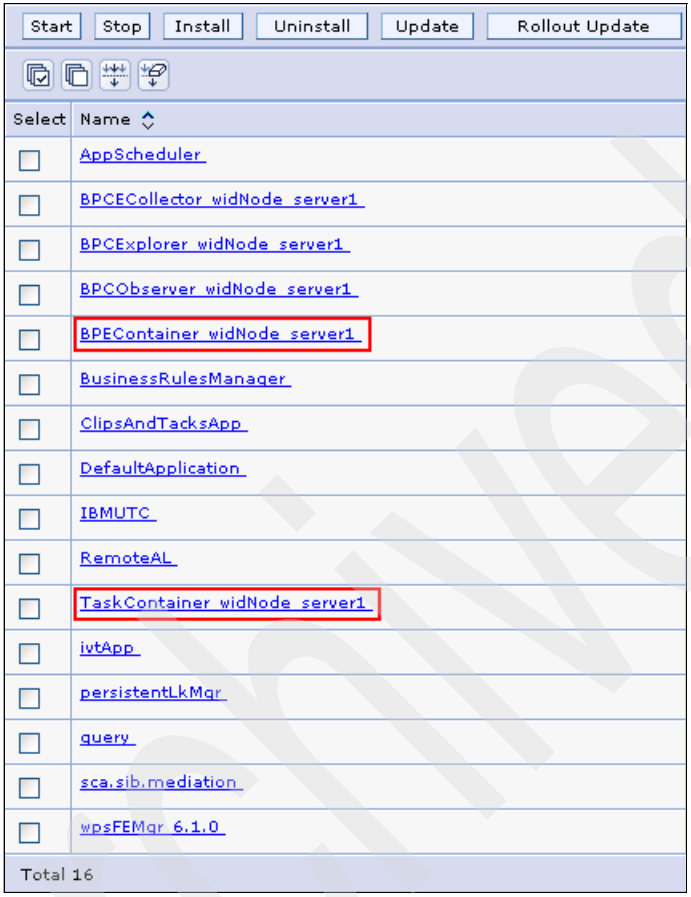


Figure A-82 Selecting the WebSphere Process Server applications

Archived

4. Under Detail Properties, click **Security role to user/group mapping** (Figure A-83).

Figure A-83 Navigating to Security role to user/group mapping for WebSphere Process Server

5. In the next panel:
   a. Select **BPESystemAdministrator** and **BPESystemMonitor** and click **Look up users**.
   b. For Search String, type an asterisk (*) as a wildcard character and click the **Search** button.
   c. From the available users listed, select **wsadmin** and click the **>** button to move the user to the selected text area. Click **OK**.
   d. For the BPEAPIUser, WebClientUser, and JMSAPIUser, select **All authenticated?** (Figure A-84).

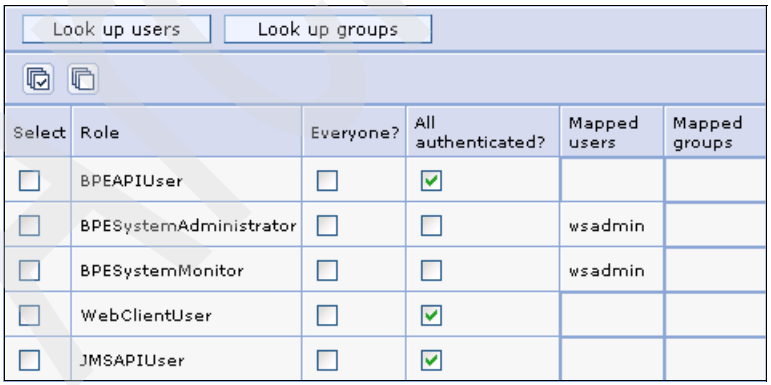


Figure A-84 User roles for BPEContainer application

   e. Click **OK**.
6. Under Detail Properties, click **User RunAs roles**.

Archived

7. Update the JMSAPIUser with the following values (Figure A-85):
   a. For username, type `wsadmin`.
   b. For password, type `itso4you`.
   c. Select **JMSAPIUser**.
   d. Click **Apply**.
   e. Click **OK**.

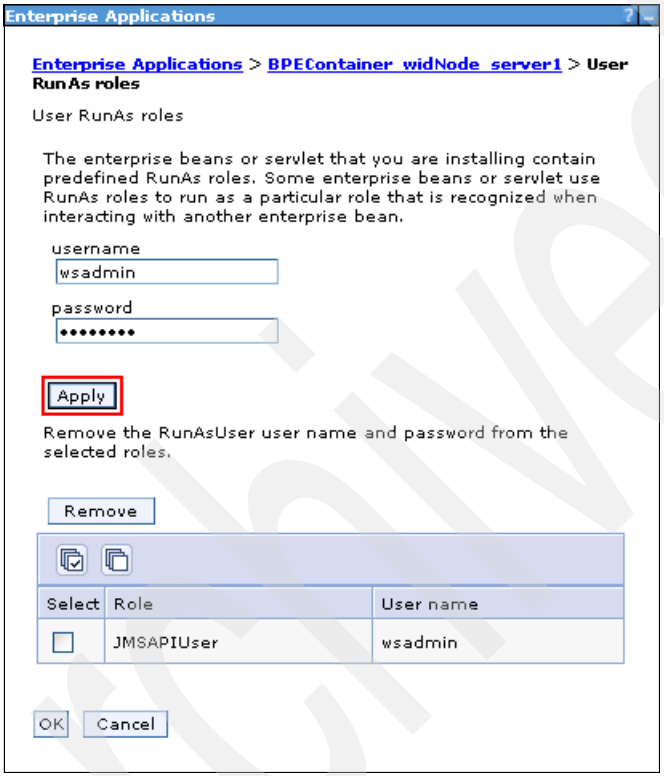


Figure A-85 Configuring JMSAPIUser

8. Save to the master configuration.
9. Repeat step 4 on page 784 through step 8 for the TaskContainer_<cellName>_server1 application (Figure A-82 on page 783). For roles, configure Task System Administrator and Task System Monitor.

Archived

Updating the J2C authentication data entries for messaging buses for WebSphere Process Server

In this section, we set up security for the messaging buses on WebSphere Process Server:

1. In the WebSphere Process Server administrative console, expand **Security** and click **Bus Security**.
2. For the BPC.widCell.Bus bus:
   a. Click the **Disabled** or **Enabled** link of the current bus (Figure A-86).

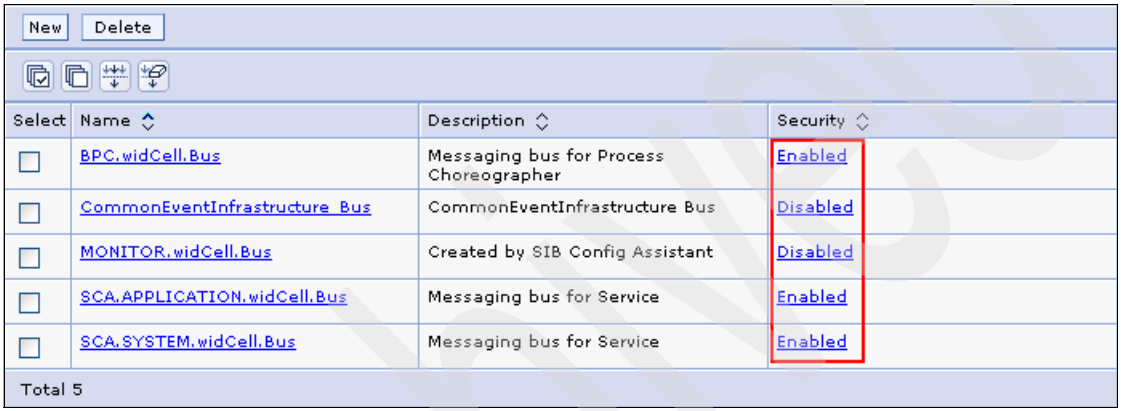


Figure A-86 Selecting the Security link of a bus for WebSphere Process Server

   b. Under Related Items, click **JAAS-J2C authentication data** (Figure A-87).

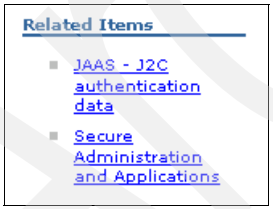


Figure A-87 Selecting JAAS-J2C authentication data for WebSphere Process Server

Archived

c. Click the **BPC_Auth_Alias** alias (Figure A-88).

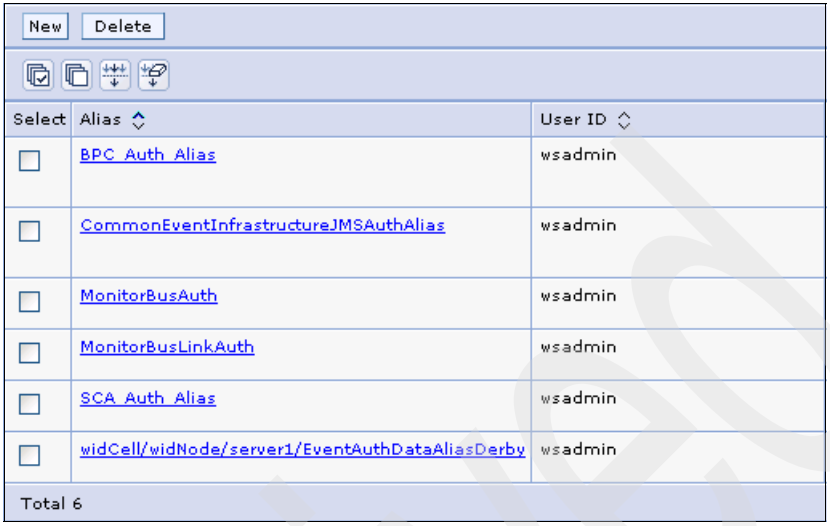


Figure A-88 WebSphere Process Server aliases

d. On the Configuration page, under General Properties (Figure A-89):

   i. For User ID, type `wsadmin`.
   ii. For Password, type `itso4you`.

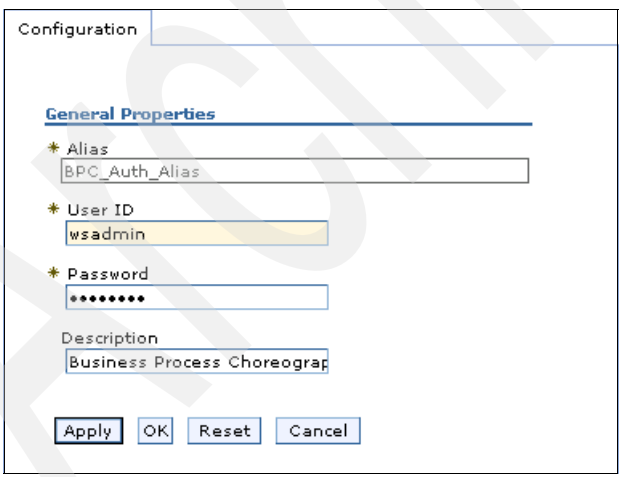


Figure A-89 Configuring aliases for the WebSphere Process Server bus

Archived

e. Repeat steps c and d on page 787 for the following aliases:

- CommonEventInfrastructureJMSAuthAlias
- MonitorBusAuth
- MonitorBusLinkAuth
- SCA_Auth_Alias
- widCell/widNode/server1/EventAuthDataAliasDerby

Alias names: The aliases may have different names in another environment.

f. Save to the master configuration.

g. Navigate back to the Disabled or Enabled link of the current bus. Under Additional Properties, click **Users and groups in the bus connector role** (Figure A-90).

Figure A-90 Navigating to Users and groups in the bus connector role for WebSphere Process Server bus

h. If an admin user exists, select the **admin user** and click the **Delete** button (Figure A-91).

i. Click the **New** button to add a new user.

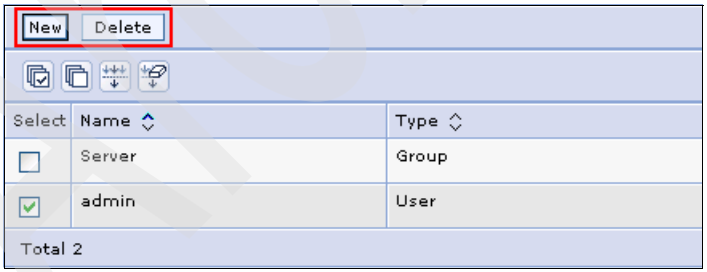


Figure A-91 Configuring the bus connector role for WebSphere Process Server

Archived

j. On the Configuration page, under General Properties (Figure A-92), select **User name** and type `wsadmin`. Click **OK**.

Figure A-92 Configuring the bus connector role for WebSphere Process Server bus

k. Save to the master configuration.

l. Navigate to the Disabled or Enabled link of the current bus.

m. On the Configuration page, under General Properties (Figure A-93):

i. Select **Enable bus security**.

ii. For Inter-engine authentication alias, type the `CommonEventInfrastructure` alias or specify any other alias since they have the same credentials.

iii. For Permitted transports, leave the defaults.

iv. Click **OK**.

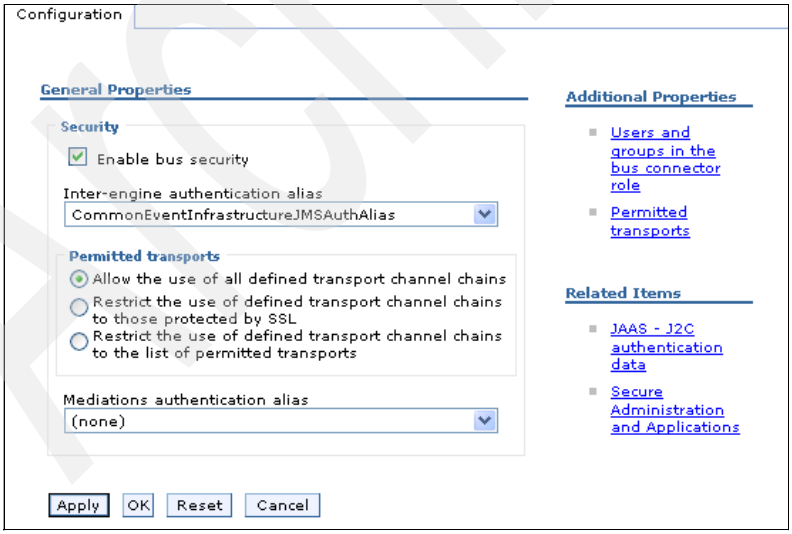


Figure A-93 Enabling WebSphere Process Server bus security

Archived

3. Repeat step 2 on page 786 for the following buses:
   - CommonEventInfrastructure_Bus
   - MONITOR.widCell.Bus
   - SCA.APPLICATION.widCell.Bus
   - SCA.SYSTEM.widCell.Bus
4. Save to the master configuration.

Granting access to the foreign queues of monitor models

WebSphere Process Server uses foreign queues to emit events to the remote WebSphere Business Monitor server. Foreign queues exist for each monitor model that is deployed on WebSphere Business Monitor. If security is enabled, the WebSphere administration user must explicitly obtain access rights to access these foreign queues:

1. In the WebSphere Business Monitor administrative console, expand **Service Integration** and click **Buses**. In the right pane, select **MONITOR.<cellname>.Bus**.
2. Under Destination resources, select **Destinations** (Figure A-94).

Figure A-94 Navigating to Destinations

Archived

3. As shown in Figure A-95, copy the name of the destination to which you want to grant the WebSphere administration user access (xxx_<Monitor Model name>_xxx_Q_Destination). In this scenario, the following destination queues are affected:
   - wbm_ClipsAndTacksMM_20080402143620_Q_Destination
   - wbm_GlobalHTMM_20070618095438_Q_Destination

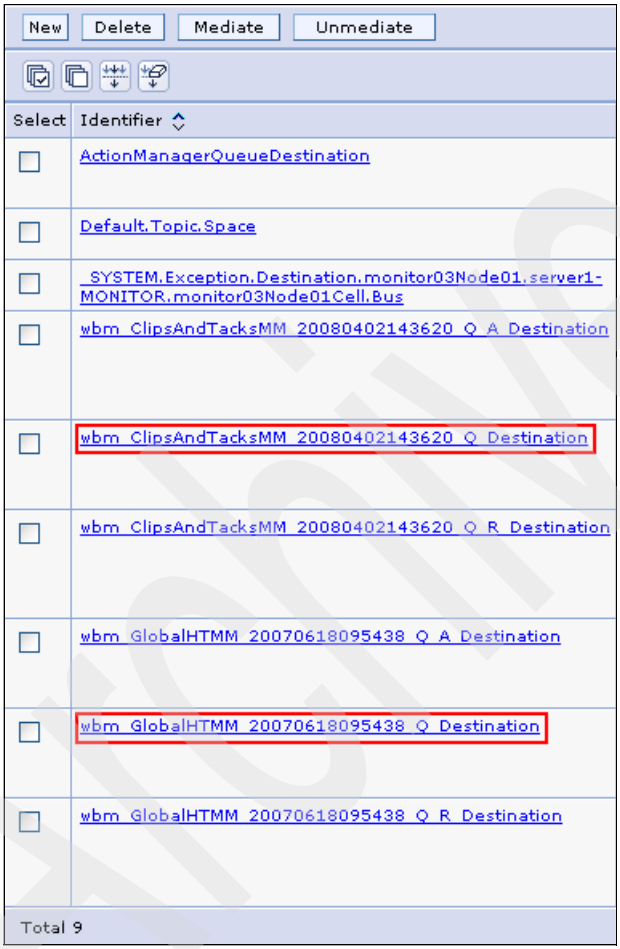


Figure A-95 WebSphere Business Monitor Monitor Bus destinations

4. On the server for WebSphere Process Server, open a command line and navigate to the bin directory of the WebSphere Process Server profile.

Archived

5. Enter the following commands for each destination to which you want to grant the administration user access (Figure A-96):

```
wsadmin.bat -username wsadmin -password itso4you

$AdminTask addUserToDestinationRole {-type foreignDestination -bus
MONITOR.<REMOTECEI_CELL_NAME>.Bus -foreignBus
MONITOR.<MonitorCellName>.Bus -destination <name of the destination>
-role Sender -user <adminUserID>}

$AdminConfig save

quit
```

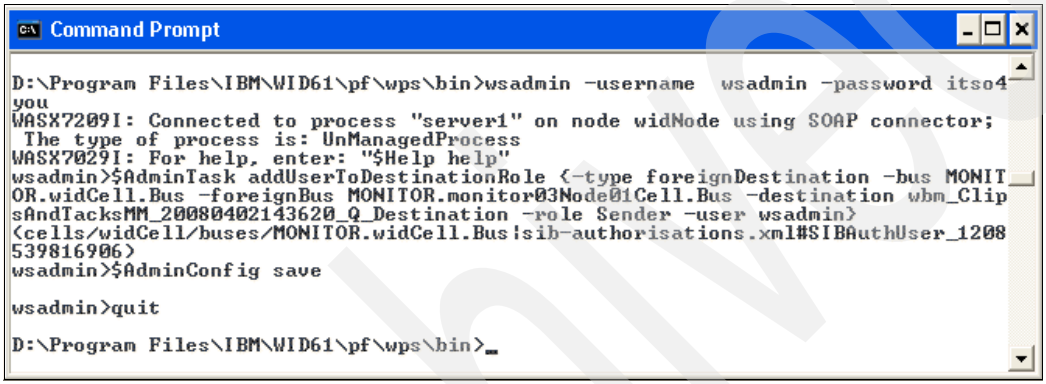


Figure A-96 Granting access rights to foreign queues

Configuring server-to-server SSL from WebSphere Business Monitor to WebSphere Process Server

In this section, we explain the server-to-server SSL setup for the environment. SSL is required to secure communications between the local and remote WebSphere Business Monitor buses.

1. Log in to the WebSphere Business Monitor server administrative console.
2. Expand **Security** and click **SSL certificate and key management** (Figure A-97).

Figure A-97 Navigating to Secure administration and infrastructure

3. Under Related Items, click **Key stores and certificates** (Figure A-98).

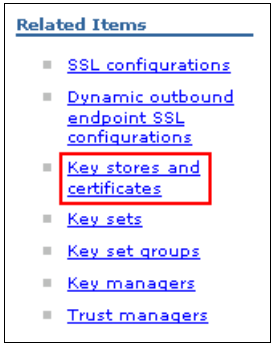


Figure A-98 Selecting Key Stores and certificates

4. Click the **NodeDefaultTrustStore** link (Figure A-99).

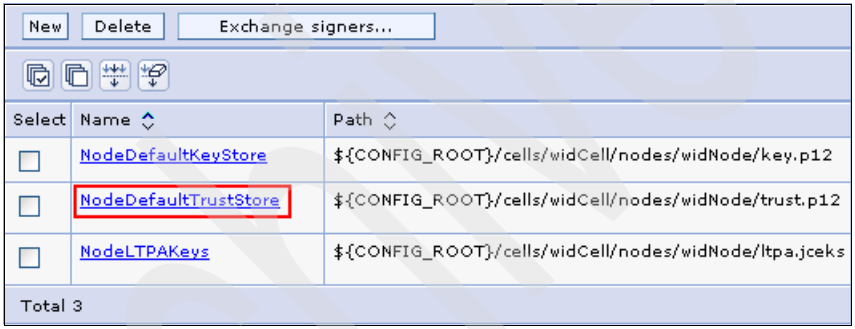


Figure A-99 Selecting DefaultTrustStore

5. Under Additional Properties, click **Signer certificates** (Figure A-100).

Figure A-100 Selecting Signer certificates

Archived

6. Click the **Retrieve from port** button (Figure A-101).

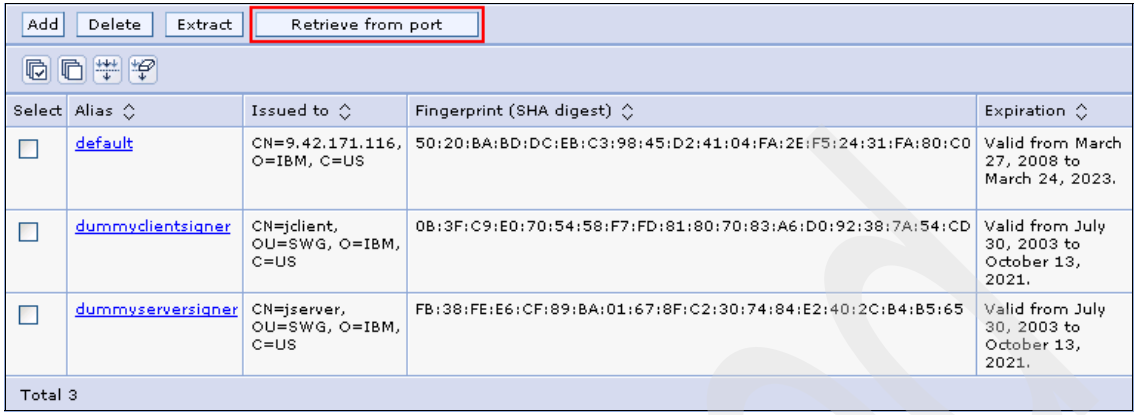


Figure A-101 Select Retrieve from port

7. On the Configuration page, under General Properties (Figure A-102 on page 795):
   a. For Host, type the fully qualified host name of the remote WebSphere Process Server CEI Server. In this example, we type `ITSO.itso.ral.com`.
   b. For Port, type the SSL port. The default is 7286.
   c. For the SSL port number:
      i. Log in to the administrative console of the remote server.
      ii. Expand **Servers** and click **Application Servers**.
      iii. Select the server, and under Communications, select **Ports**.
      iv. Check the port number for SIB_ENDPOINT_SECURE_ADDRESS.
   d. For Alias, type `remoteCEI`.
   e. Click the **Retrieve signer information** button. After you click the Retrieve signer information button, the signer information is displayed as shown in the lower half of Figure A-102 on page 795.
   f. Click **OK**.

Archived

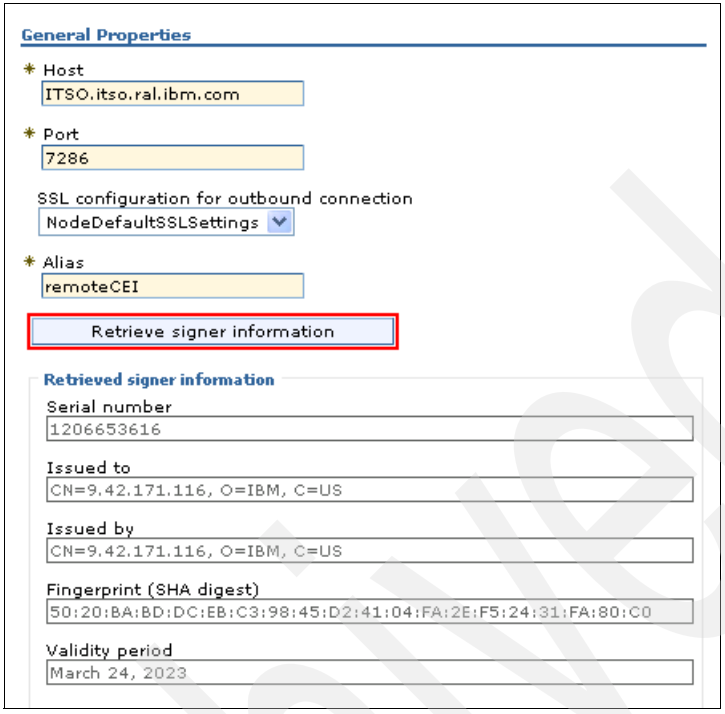


Figure A-102 Retrieve Signer information

8. Save to the master configuration.
9. Repeat the previous steps for the WebSphere Process Server.
10. Restart both servers and check the SystemOut.log files for security related error messages.

Configuring the dashboard for human task monitoring

In this section, we set up a Human Tasks view in the WebSphere Business Monitor Web-based dashboard. We also explain the advanced features of this view.

Archived

Creating a Human Tasks view

To create a Human Tasks view:

1. Use the Business Process Choreographer Explorer to execute process instances involving human tasks as explained in “Starting process instances” on page 230.
2. Launch the WebSphere Business Monitor Business dashboard.

> **BusinessDashboard:** You can access the BusinessDashboard of WebSphere Business Monitor by typing the following URL:
>
> `http://<host name>:<WC_defaulthost port>/BusinessDashboard`

1. Log in by typing `wsadmin` for User ID and `itso4you` for Password.
2. Select the **Dashboards** tab and click the **New** button.
3. Create a new dashboard:
   a. Enter a name and click **OK**.
   b. Click the **Add to Dashboard** link.
   c. Select **Human Tasks** and click **OK**.
4. In the Human Tasks panel (Figure A-103):
   a. Click the **Personalize** button.
   b. Click the **>>** button to select all available fields or click the **>** button to choose selected fields.
   c. Click **OK**.

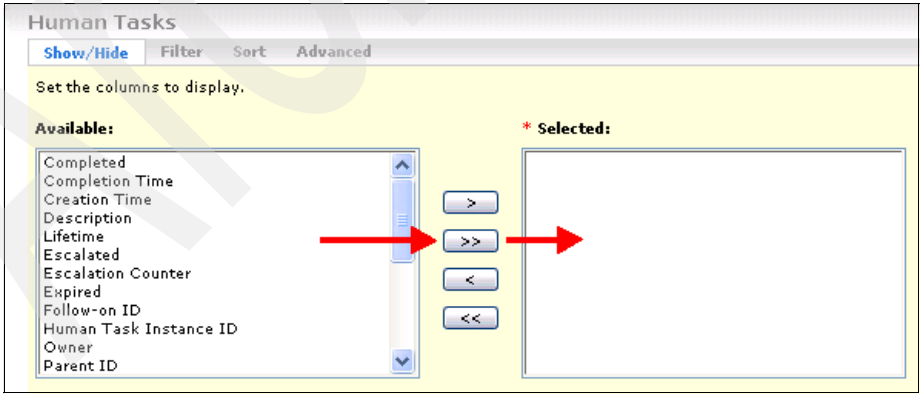


Figure A-103 Selecting the fields for human tasks

Archive

The Human Tasks view shows all executed human tasks (Figure A-104).

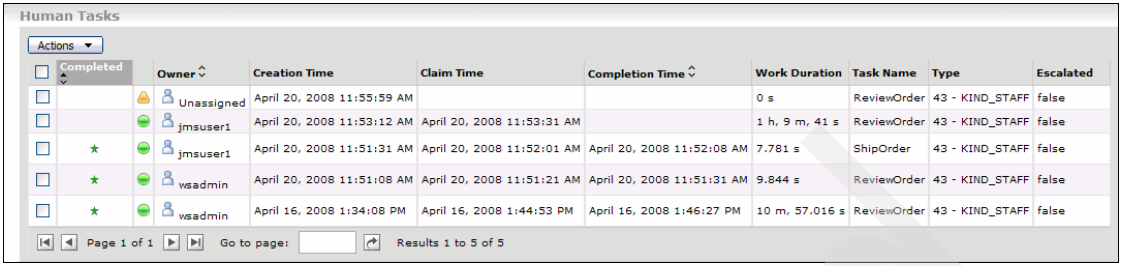


Figure A-104 Human Tasks view

Advanced features of the Human Tasks view

The Human Tasks view of the WebSphere Business Monitor Web-based dashboard provides advanced human task management features. You can find all available features by clicking the **Action** button in the Human Tasks view (Figure A-105).

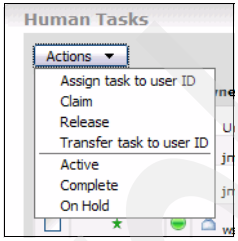


Figure A-105 Actions of the Human Tasks view

Claiming a human task

To claim a human task:

1. Select an unassigned human task in the task list.
2. Click the **Actions** button and select **Claim** (Figure A-106).

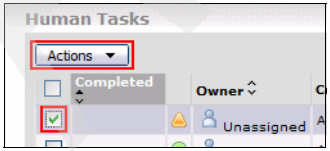


Figure A-106 Selecting an unassigned human task

Archive

3. In the message window (Figure A-107) that indicates that the claim action was successful, click **OK**.

Figure A-107 Successful action message

As shown in Figure A-108, the human task is assigned to the logged in user after the next refresh.

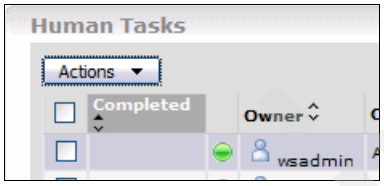


Figure A-108 Assigned human task

Releasing a human task

To release a human task:

1. Select an incomplete human task assigned to the logged in user.
2. Click the **Actions** button and select **Release** (Figure A-109).

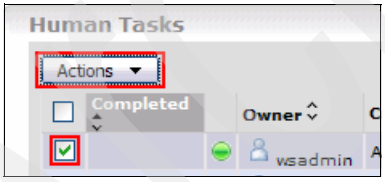


Figure A-109 Selecting an unassigned human task

3. In the message window that indicates a successful release action, click **OK**.

As shown in Figure A-110, the human task is marked as *Unassigned* afterwards.

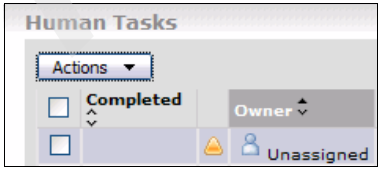


Figure A-110 Unassigned human task

Archived

Transferring a human task to a user ID

To transfer a human task to a user ID:

1. Select an incomplete human task assigned to the logged in user.
2. Click the **Actions** button and select **Transfer task to user ID** (Figure A-111).

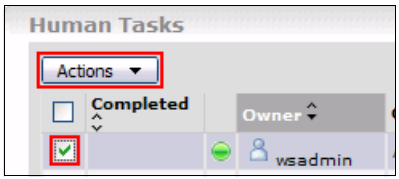


Figure A-111 Selecting an incomplete assigned human task

3. In the Human Tasks panel (Figure A-112), type the user ID to which the human task should be transferred and click the **Transfer** button.

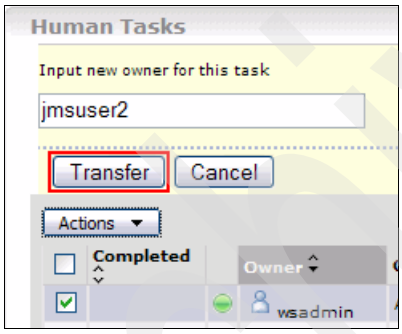


Figure A-112 Providing a user ID to transfer a human task

4. In the message window that indicates a successful transfer action, click **OK**.

As shown in Figure A-113, the human task is assigned to the chosen user.

Figure A-113 Transferred human task

Archived

Assigning a human task to a user ID

Note: This feature is supported only if the staff expression of the human task, defined in the BPEL process, is not set to *Everybody*. Otherwise every user automatically receives a work item for this task, and there is no user left to whom the task can be assigned.

To assign a human task to a user ID:

1. Select an unassigned human task in the task list and click the **Actions** button.
2. Select **Assign task to user ID**.
3. Enter the user ID to which the human task should be assigned and click the **Assign** button.

Working on a human task

The WebSphere Business Monitor dashboard enables the possibility to work on human tasks. The business user can perform the following tasks:

- Complete a human task:
  a. Select an incomplete human task claimed by the logged in user.
  b. Click the **Action** button and select **Complete** to complete the human task.
- Place a human task on hold:
  a. Select an incomplete human task.
  b. Click the **Action** button and select **Hold** to set the human task on hold and set it in a *Suspend* status.
- Activate a human task:
  a. Select a suspended or on hold human task.
  b. Click the **Action** button and select **Active** to activate the human task.

Summary

In this appendix, we explained how to monitor WebSphere Process Server human tasks. We provided all steps for this type of monitoring including the installation of a Global Human Task monitor model and security setup for the WebSphere Process Server - WebSphere Business Monitor environment. In the last section, we explained how to create a Human Tasks view on the dashboard and provided information about advanced features for this kind of dashboard component.

Archived

B

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 the following address:

ftp://www.redbooks.ibm.com/redbooks/SG247638

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, SG247638.

Archived

© Copyright IBM Corp. 2008. All rights reserved.

Using the Web material

The additional Web material that accompanies this book includes the following files:

File name	*Description*
sg247638.zip	Compressed code samples

How to use the Web material

Create a subdirectory (folder) on your workstation, and extract the contents of the Web material compressed file into this folder.

Archived

Abbreviations and acronyms

API	application programming interface
BAM	business activity monitoring
BPC	Business Process Choreographer
BPEL	Business Process Execution Language
BPM	business process management
CEI	Common Event Infrastructure
CMP	container managed persistence
CSDL	China Software Development Lab
EAR	enterprise archive
ECM	Enterprise Content Management
EIS	Enterprise Information System
ESB	enterprise service bus
FDL	Flow Definition Language
FTP	File Transfer Protocol
GUI	graphical user interface
IBM	International Business Machines Corporation
IDE	integrated development environment
ITCAM	IBM Tivoli Composite Application Management
ITSO	International Technical Support Organization
J2EE	Java 2 Enterprise Edition
JCA	J2EE Connector Architecture
JMS	Java Messaging Service
JNDI	Java Naming and Directory Interface
JSON	JavaScript Object Notation
KPI	key performance indicator
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Directory Interchange Format
LOB	large object
MC	monitoring context
POJO	Plain Old Java Object
REST	Representational State Transfer
RHEL	Red Hat Enterprise Linux
RMI	Remote Method Invocation
SCA	Service Component Architecture
SDO	Service Data Object
SIB	service integration bus
SLES	SUSE Enterprise Linux Server
SMO	Service Message Object
SOA	service-oriented architecture
SSL	Secure Sockets Layer
SVG	Scalable Vector Graphics
TPTP	Test and Performance Tools Platform
XSD	XML Schema Definition

Archive

© Copyright IBM Corp. 2008. All rights reserved.

Archived

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 806. Note that some of the documents referenced here may be available in softcopy only.

- *Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products*, SG24-7148
- *Getting Started with WebSphere Process Server and WebSphere Enterprise Service Bus Part 1: Development*, SG24-7608
- *Getting Started with WebSphere Process Server and WebSphere Enterprise Service Bus Part 2: Scenario*, SG24-7642

Online resources

These Web sites are also relevant as further information sources:

- Architecture Technical library view in IBM developerWorks®

  http://www.ibm.com/developerworks/views/architecture/libraryview.jsp?search_by=put+new+%20capabilities+of+bam+to+work

- Automic computing

  http://www.ibm.com/autonomic

- IA9V: IBM WebSphere Message Broker - Common Base Event Publisher

  http://www-1.ibm.com/support/docview.wss?rs=977&context=SSKMAB&context=SS7J6S&context=SSKM8N&context=SSFKSJ&context=SSFKUX&context=SSWHKB&context=SSVLA5&dc=D410&q1=SupportPac+IA9V&uid=swg24016414&loc=en_US&cs=utf-8&lang=en

- IBM SOA Business Catalog home page

  http://catalog.lotus.com/wps/portal/soa

© Copyright IBM Corp. 2008. All rights reserved.

- WebSphere Application Server information center

  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp
- WebSphere Business Monitor V6.1 Information Center

  http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp
- WebSphere Business Monitor system requirements

  http://www-306.ibm.com/software/integration/wbimonitor/requirements/
- “What’s new in WebSphere Business Monitor 6.1”

  http://www.ibm.com/developerworks/architecture/library/i-bam611/

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Archived

Index

Symbols

${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH} 282
${DB2UNIVERSAL_JDBC_DRIVER_PATH} 282

A

acquisition 140
action manager 225
action services 27
activation specification 770
Active Content management 353
ad hoc event generation 729
Add Base entry to Realm button 752, 775
Add Repository button 753
administrative security 83
Advanced Human Tasks view features 797
advanced installation 52
Aggregate KPI 135
AIX 5L 71
Ajax 46
alert creation 204
Alert Subscription utility 33
Alert User IDs 770
Alerts view 30
  creation 242
alias queue 686
AlphabloxAdministrator 761
AlphabloxDeveloper 762
AlphabloxPlatform 760
AlphabloxUser 762
API (application programming interface) 483
application programming interface (API) 483
Application Security Warnings page 98
application server profile, stand-alone 52
ApplicationStudio 762
ApproveApprove 416
approver 397
arbitrary events 348
archive folder 610
assembly diagram 159
assured event delivery 661
asynchronous beans 359
asynchronous processing 40
attachments 396
audit information 145
augmentation
  deployment manager profile 52
  existing monitor model 328
authentication mechanisms and expiration 758
availability 5
available realm definitions 751

B

BAM (business activity monitoring) 4
  components 6
    business activity 6
    business event 6
    KPI 6
    metric 6
base entry for default parent values 755
bind distinguished name 753
bind password 753
BOOTSTRAP_ADDRESS 615
bottlenecks 12
BPC.widCell.Bus 786
BPESystemMonitor 784
BPM Business experts 5
BPMN 353
BPMSTATE database 358
built-in component-generic events 254
built-in component-specific events 254
bulk processing 145
bus members 680
bus security 764
business activity 6
business activity monitoring (BAM) 4
  components 6
    business activity 6
    business event 6
    KPI 6
    metric 6
business customer 145
business event 6
business executives 5
business integration 150, 260
business intelligence 13

© Copyright IBM Corp. 2008. All rights reserved.

business leader 5
business monitoring perspective 14
business object maps 151
business process management 140
business processes 151
business rules engine 143
business situation events 225
business state machines 152
business-manager 771
business-related data 13

C

CEI (Common Event Infrastructure) 6, 150, 251
  changing distribution 107
  components 7
    Common Base Event 7
    emitter 7
    event catalog 8
    event consumer 8
    event service 7
    event source 8
  distribution mode 104
  event definition 162
CEIMQ 683
certificate mapping 753
CICS 669
cleaning all projects 327
ClipsAndTacks 140
  contract management process 141
  customer credit check service 142
  external shipping service 142
  new division order handling process 141
  order handling process 141
  order processing database 142
CMP (container managed persistence) 371
COBOL 148
  copybooks 148
command server for remote administration 678
Common Base Event 7
  adapter 354
  adapter database 357
  browser 293
  model 8
  situation 9
common business event 152
Common Event Infrastructure (CEI) 6, 150, 251
  changing distribution 107
  components 7
    Common Base Event 7
    emitter 7
    event catalog 8
    event consumer 8
    event service 7
    event source 8
  distribution mode 104
  event definition 162
CommonBaseEvent 9
CommonEventInfrastructure_ActivationSpec 770
CommonEventInfrastructure_AllEventsTopicCF 769
CommonEventInfrastructure_Bus 679, 790
competitive pressures 140
component authentication alias 769
configCEIForMQClients.bat 683
configRemoteMonitorBus.sh 220
connection management 661
container managed persistence (CMP) 371
content completion 605
content handler 604
ContentHandler 604
continuous process optimization 4
contractual relationships 145
cost cutting 4
counter 19
  creation 198
CreateSituation 9
cross-cell single sign-on 758
cube 21
custom Java event emitters 43
custom mediation primitive 319
custom properties 379

D

dashboard 4
  alert 458
  creation 235
  KPI 50, 128
  queries 27
DashboardBPCHost 746
DashboardBPCRMIPort 746
Data Access Builder 39
Data Fields tab 392
data mart model 387
  cube for process 387
  cube for work item 387
  cubes 387

dimension 387
Data Movement Service node 91
data movement services 27
data security for monitor models 771
data source for the scheduler 633
DataPower XI50 42
day-to-day operations 49
DB2 Alphablox 25
DB2 Cube Views 52
DB2 Replication Manager 27
DB2 Universal Database 53
DB2 Universal JDBC Driver Provider (XA) 284
db2jcc_licence_cu.jar 264
db2jcc.jar 264
delete schema scripts 113
deploying monitor models 90
deployment manager profile 52
Derby Embedded database 37, 58
destinations 667
development test environments 57
development toolkit 13
development topology 56
developmentMode=true 37
Diagrams view 30
creation 238
dimension 21
creation 203
dimensional model 21
Dimensions view 32
creation 242
distribution mode 100
Dojo 46

E

EmissionController 608
emitter 7, 605
factory 605
framework 601
emitterFactoryJNDI 615
EmitterFormatterFactory 602
enterprise applications 782
enterprise management 8
ENTRY 253
ESB 249
event 9
catalog 8, 254
class 9
consumer 8
definition 162
emitter 670
formatter 616
generation 291, 327
group 18
logs 392
model 22
monitoring directly from the event sources 347
nature 253
payload 5
point 253
sequencing 40
service 7
service security 287
source 8
transmission 605
trigger 186
event content 163
digest 163
empty 163
full 163
Event Emitter primitive 320
event emitter scheduler 637
Event Monitor tab 163, 740
EventFactory 604
EventFileBean 608
EventFolderPoller 608
EventFormatter 602
EXACT_DN 753
existing projects in workspace 610
EXIT 253
Export Delete Schema Script 114
Export Keys button 759
Export Values utility 33
Expression KPI 135
expression snippet 322
extendedDataElements 40
extensible 9
extensionName 9

F

failover 57
FAILURE 253
fast path options 37
fault folder 610
FDL (Flow Definition Language) 482
to monitor model utility for WebSphere MQ Workflow 483

Archived

federated repositories 749
FileNet P8 BPM 352
FileNet P8 BPM CBE events
  activity events 383
    activity 383
    activityId 383
    boundUser 383
    enqueueTime 383
    map 383
    operation 383
    P8.BPM.Activity.Failure 383
    P8.BPM.Activity.Locked 383
    P8.BPM.Activity.Start 383
    P8.BPM.Activity.Unlocked 383
    P8.BPM.WorkItem.End 383
    queue 383
  common extended data 382
    businessData 382
    isolatedRegion 382
    logName 382
    processName 382
    processStartTime 382
    processVersion 382
    username 382
  correlation data 382
    parentId 382
    processId 382
    workItemId 382
  header information 381
    creationTime 381
    extensionName 381
    sequenceNumber 381
    version 381
  process events 382
    P8.BPM.Process.End 382
    P8.BPM.Process.Start 382
  work item events 382
    P8.BPM.WorkItem.End 382
    P8.BPM.WorkItem.Start 382
FileNet P8 BPM Monitor Model for WebSphere 478
FileNet P8 BPM Monitoring Dashboard for WebSphere Business Monitor 479
FileNet Workplace 388
First Steps window 52
flexible topologies 666
Flow Definition Language (FDL) 482
  to monitor model utility for WebSphere MQ Workflow 483
FMC 492
FMC_MQAudit_FMCGRP 503
FMC.FMCGRP.AUDITQ 494
fmczchk 492
foreign bus definition 686
foreign queues of monitor models 790

G

Gartner 4
Generate Keys button 758
Generate Monitor J2EE Projects window 224
Get Data button 584
Getting Started guide 45
Global Event Settings tab 164, 292
Global Human Task monitor model 50, 741
global instance identifier 605
GlobalHTMMApplication.ear 743
government 4
group 756, 779
group attribute definition 756
growable 360

H

high availability capabilities 57
hosts file 495
HP-UX Itanium 53
HTTP services 27
human task 115, 151
  assignment to a user ID 800
  claiming 797
  monitoring 795
  releasing 798
  tranferring to a user ID 799
  working on 800
Human Tasks view 33, 796

I

IBM Autonomic Computing Architecture Board 8
IBM Autonomic Computing Toolkit 9
IBM enterprise content management solutions 352
IBM FileNet P8 Enterprise Content Management 352
IBM Installation Manager 67
IBM Tivoli Composite Application Management (ITCAM) for SOA 43
IBM Tivoli Directory Server V6.0 738
IBM WebSphere MQ Workflow Configuration Utility 492

IBM_WBM_REST_SERVICES 763
IBM_WBM_WEB_DASHBOARD 763
implementation generation 314
inbound event 18
inbound event filter 478
inbound folder 610
IndexComplete 411
indexer 145, 397
indexing 410
information center 73
Informix 366
input data structure 583
inserts 27
Instances view 29
  creation 236
INSTCONFSUCCESS 68
Integrated Test Client 15, 23
Inter-engine authentication alias 768
interface maps 151
IT experts 5
ITCAM (IBM Tivoli Composite Application Management) 43
ITSO Logistics 147

J

J2C Authentication Alias 278
J2EE (Java 2 Enterprise Edition) 250
J2EE Connector Architecture (JCA) 44
J2EE Security 661
JAAS (Java Authentication and Authorization Service) 278
Java 2 Enterprise Edition (J2EE) 250
Java 5 annotations 44
Java and process management 378
Java Authentication and Authorization Service (JAAS) 278
Java Build Path 612
Java Database Connectivity (JDBC) 44
Java event emitter interface 43
Java Messaging Service (JMS) 666
Java virtual machine 378
JavaScript Object Notation 38
JCA (J2EE Connector Architecture) 44
JDBC (Java Database Connectivity) 44
JDBC Data Source 283
JDBC Provider 280
JDBC Provider for the scheduler 629
JMS (Java Messaging Service) 666
JMSAPIUser 785
JSON 38

K

key 19, 648
key file name 759
key performance indicator (KPI) 21
  context 21
  creation 198
  model 21
  modeled 49, 128
  visualization 46
key stores and certificates 793
KPI (key performance indicator) 6, 21, 198
  context 21
  model 21
  modeled 49, 128
  visualization 46
KPI Manager 49, 128
  Add user 130
  Aggregate KPI 135
  Copy a KPI 135
  Delete a KPI 135
  Expression KPI 135
  KPI-Administrator 129
  modeled KPI 132
  Modify 132
  overview 131
  Personal-KPI-Administrator 129
  Public-KPI-Administrator 129
  Utilities tab 132
KPI Manager utility 33
KPI-Administrator 771
KPIs view 29

L

launchpad 64
LDAP Directory Interchange Format (LDIF) 738
LDAP entity types 756
LDAP server 52
LDAPBindID 738
LDAPBindPassword 738
LDAPHostName 738
LDAPPort 738
LDAPServerType 738
LDAPSuffix 738
LDIF (LDAP Directory Interchange Format) 738
Lightweight Directory Access Protocol (LDAP) regis-

Archived

try 738
listener configured for TCP/IP 676
local file system 681
Look up users button 761

M

managed process 142
map
  expressions 44
  security roles to users or groups 287
  WBI.MAP.ENTRY 254
  WBI.MAP.EXIT 254
  WBI.MAP.FAILURE 254
  WBI.MAP.Transformation.ENTRY 254
  WBI.MAP.Transformation.EXIT 254
  WBI.MAP.Transformation.FAILURE 254
maxEventsPerSecond 379
measure 22
measure business performance 142
measurement variables 13
mediation flow 151, 315
  component 312
  custom events 254
  editor 315
mediation primitive 312
  stopping 324
member attributes 757
merging with existing monitor model 330
message channels 666
message filter primitive 315
message parameters 666
message routing 666
message-oriented middleware 666
messaging engines 221
metric 6, 18
  creation 188
Microsoft Active Directory Server 355
Microsoft SQL Enterprise 2000 355
Migration wizard 54
milestones 396
modeled KPI 49, 128, 132
modularity 661
mon61to61CrossCell.zip 220
Monitor Action Services 226, 458
  node 91
Monitor Application Descriptor models 63
Monitor Data Security 130, 771
Monitor database 27
monitor details model 18
Monitor development environment 63
monitor model 13
  events editor 23
  generation 170
  manager 26
  merging with existing monitor model 330
  purging 112
  templates 34
  terminology 18
  versioning 104
Monitor Model editor 14, 17
Monitor Models node 90
Monitor server
  starting 89
  stopping 89
Monitor test environment 63
MONITOR.widCell.Bus 790
MonitorAlphabloxAlias 766
MonitorBusAuth 767, 769
MonitorBusLinkAuth 767
Monitoring arbitrary events 600
monitoring context 18, 27
MonitorQueueConnectionFactoryAuth 767
moveMCInstances command 107
MQ clustering 487
MQ Link 671
MQ to CEI bridge 678
MQLink 686
MQtoCEIMediation 682
multi-version environment 37

N

NameNotFoundException errors 346
native library path 374
network latencies 40
new process template list creation 582
nonmanaged process 142
non-persistent messaging 666
notifications 226, 458, 770

O

ongoing performance 148
operation connections 315
operational tables 27
Oracle 27
Oracle 10g Enterprise 53
Oracle 10g Standard 53

Archived

organizations 33
OrgContainer 756, 779
osgiCfgInit.sh 220
outbound event 19
output terminal 316

P

P8BPMCBEAdaptor.ddl script 358
page creation interface 50
PendSendToApproval 414
performance 5
persistent messaging 666
Person Finder portlet 51
PersonAccount 756, 779
personalization 46
Personal-KPI-Administrator 771
point-to-point communication 666
Portal dashboard 26
portlet-based dashboard 24
print to log 321
process
  definition 378
  designer 391
  instance list 582
  level context 384
  template lists 582
  users 5
process engine 354
  database 366
  software 479
process monitor context 384
  inbound events 385
  key 384
  metrics 385
  stopwatch 385
  triggers 385
processing time stopwatch 385
programmatic custom events 255
project cleaning 327
proof-of-concept environments 57
public inboxes 410–411
Public-KPI-Administrator 771
publish-and-subscribe 353
purging a monitor model 112

Q

queue manager 672
  clustering 666

R

Rational Application Developer 14
receiver channel 686
Red Hat Enterprise Linux 53
Redbooks Web site 806
  Contact us xvii
refactoring 34–35
relationships 151
Relative Distinguished Name properties 755
remote administration channel 678
remote CEI 218
Remote Method Invocation (RMI) 84
remote queue definition 686
reporting tables 27
Reports view 31
  creation 244
repository identifier 753
Representational State Transfer (REST) 44
  APIs 27
  service 26
republish support 63
REST (Representational State Transfer) 44
  APIs 27
  service 26
Retrieve from port button 794
Retrieve signer information button 794
reusability 662
reviewer 145, 397
ReviewPend 412
RMI (Remote Method Invocation) 84
role-based information 4
run delete schema script 114

S

SAP 42
SCA (Service Component Architecture) 44, 50
SCA.APPLICATION.widCell.Bus 790
SCA.SYSTEM.widCell.Bus 790
scalability 661
scheduler for the event emitter 637
schema deletion during installation script 111
Script to delete the schema during uninstallation 111
SDP70Shared 510
security
  management 661
  WebSphere Business Monitor 747
  WebSphere Process Server 773

Archive

WebSphere Process Server applications 782
Security Configuration wizard 747
security role to user/group mapping 761
seed monitor model 297
selectors 151
sender channel 686
sequence number 605
server-to-server SSL 792
Service Component Architecture (SCA) 44, 50
service component monitoring 253
service message object (SMO) 251
shipping dock 147
signer certificates 793
single server topology 57
situation 9
situation event bindings 228
SMO (service message object) 251
SOAP_CONNECTOR_ADDRESS 220
software updates 511
special business situations 12
SQL server 366
SSL 792
certificate and key management 792
staff workload 4
stand-alone application server profile 52
standard visual snippet 321
start instance 230
StartSituation 9
statsInterval 379
stopping the mediation primitive 324
stopwatch 19
creation 178
supervisor 397
supported entity types 755
SupportPac IA9V 42
SUSE Linux Enterprise Server 53
SVG file 22
Sybase 366
synchronization features 34–35
system administration functions 252

T

technical monitoring 5
template definitions 226, 458
TerminationTime 467
test script editor 23
thread pool 359
three-server distribution 59
topic connection factories 769
touch points 142
traditional business approach 140
transaction 164
default 164
existing 164
management 661
new 164
transformation element 254
transmission queue 686
trend analysis 12
trigger 19, 44, 649
creation 178
troubleshooting 640
two-server distribution 59

U

unauthorized business users 50
unstructured content 145
update installer 52
updates 27
User RunAs roles 784
user-defined XPath functions 44
users and groups in the bus connector role 767

V

versioning 104
monitor model 104
visual model 22
creation 210
Visual Model editor 35
visualizations with Web 2.0 46

W

WAS_61_PLUGINDIR 612
Web interface 141
Web services 396
Web-based dashboard 24, 115
adding items 120
alerts 115
alerts subscription 116
bar graph 123
dashboard creation 117
Dashboards tab 119
diagrams 115
dimensions 115
export values 116

Archived

full gauge 123
Getting Started guide 118
grid 124
half gauge 123
horizontal 124
Human Tasks 115
instances 115
KPI 115
KPI Manager 116
launch 117
Manage tab 119
personalization 121
personalize mode 116
reports 115
vertical 124
view mode 116
WebSphere Adapter for JDBC 262
WebSphere Application Server 52, 150
Network Deployment 150
WebSphere Business Integration
Adapter framework 250
Adapters 250
WebSphere Business Monitor
architecture 13
buildtime architecture 14
dashboards 28
development toolkit 14, 62
installation preparation utility 71
installation verification 88
new in V6.1 34
adding and changing KPIs 49
advanced integration with WebSphere Portal 50
alerts and actions 50
authoring WebSphere Business Monitor models 34
developing custom BAM interfaces 38
First steps 53
Getting Started guide 45
installation scenarios 52
integrated dashboards 37
integrated test client 37
monitoring business applications 42
monitoring business processes 41
monitoring other business activity 42
operating environments 53
pulling and augmenting business data with user-defined functions 44
role-based dashboard access 50
unit test server 36
unit testing with integrated Monitor server 36
visualizations with Web 2.0 46
WebSphere DataPower Integration Appliance XI50 42
WebSphere Enterprise Service Bus and WebSphere Adapters 42
prerequisites 71
profile 65
runtime architecture 24
security 747
server 26, 63
server configurations 52
test environment 22
WebSphere Business Monitor Web dashboard 70
WebSphere DataPower Integration Appliance XI50 42
WebSphere embedded ConnectJDBC driver for MS SQL Server 370
WebSphere Enterprise Service Bus 249
with Adapters 661
WebSphere Integration Developer 14
WebSphere MQ 488, 665
WebSphere MQ audit 508
event codes 508
21000 508
21006 508
21007 508
21011 508
21015 508
21016 508
21017 508
21018 508
21020 508
21025 508
21026 508
activity ended normally 508
activity ready 508
activity started 508
block ended 508
block started 508
process created 508
process created and started 508
process deleted 508
process ended normally and deleted 508
process started 508
process terminated and deleted 508
WebSphere MQ Client channel 488

WebSphere MQ Explorer 672
WebSphere MQ Workflow 481
  buildtime architecture 482
  event converter 485
  execution server 482
  lient 581
  process deployment 575
  ServicePack 6 484
  SupportPac WA61 485, 494
WebSphere Portlet Factory 39
WebSphere Process Server
  application security 782
  integration into WebSphere Business Monitor 153
  security 773
  Version 6 150
WebSphere profile management 52
welcome page 610
wf360w6a.exe 490
Windows Server 2003 71
Windows XP Professional 71
WMQWF folder 509
WMQWF/eclipse 516
work item level context 384
work item monitor context 386
  inbound events 386
  key 386
  metrics 386
  stopwatch 386
  triggers 386
work manager 358–359
workflow groups 396
worklist 582
world clock portlet 51
WS-Notification 43

X

XML Process Definition Language (XPDL) 353
XML Schema Definition (XSD)
  definitions 662
  schema 9
XPath 44
XPath Expression Builder 324
XPDL (XML Process Definition Language) 353
XSD (XML Schema Definition)
  definitions 662
  schema 9

Z

z/OS 53

Archived

Redbooks

Business Activity Monitoring with WebSphere Business Monitor V6.1

(1.5” spine)
1.5”<-> 1.998”
789 <->1051 pages

Redbooks

Archived

Archived

Business Activity Monitoring with WebSphere Business Monitor V6.1

Monitor processes in WebSphere Process Server, MQ Workflow, and FileNet

Monitor WebSphere MQ and WebSphere Adapters

Learn by example with practical scenarios

Business activity monitoring (BAM) is a solution that provides real-time tracking of business events, including the tracking of business processes, operational activities, and business situations. The business events that drive business activity monitoring can be sent by a variety of applications and technologies. The collected event information is available in the form of dashboards.

Through step-by-step hands-on examples, in this IBM Redbooks publication, we explore how you can use IBM WebSphere Business Monitor V6.1 to track business events from a variety of applications. We discuss combining WebSphere Business Monitor with process-based applications running in IBM WebSphere Process Server, WebSphere MQ Workflow, and FileNet P8 BPM to achieve business activity monitoring.

In addition, we explore monitoring business events from non-process based applications including WebSphere MQ, and WebSphere Adapters running within WebSphere Enterprise Service Bus (ESB). We also explore how to create a custom event emitter.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7638-00 ISBN 0738431133