

Implementing an Advanced Application

Using Processes, Rules, Events, and Reports

Defining processes with Business Process Manager and WebSphere Operational Decision Management

Data modeling with InfoSphere Data Architect

Reporting with Cognos Business Intelligence



Ahmed Abdel-Gayed
Kulvir Singh Bhogal
Don Carr
Richard Davies
Aditya P Dutta
Marcelo Correia Lima
Agueda Martinez Hernandez Magro
Yuka Musashi
Michael Norris
Felix Pistorius



International Technical Support Organization

**Implementing an Advanced Application Using
Processes, Rules, Events, and Reports**

October 2012

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

First Edition (October 2012)

This edition applies to Version 8 of IBM Business Process Manager and related products.

© Copyright International Business Machines Corporation 2012. 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	vii
Trademarks	viii
Preface	ix
The team who wrote this book	ix
Now you can become a published author, too!	xi
Comments welcome	xi
Stay connected to IBM Redbooks	xii
Chapter 1. Introduction to the solution and the scenario used in this book	1
1.1 Business scenario summary	2
1.2 The solution architecture used in this book	2
1.2.1 IBM products used in the solution	3
1.2.2 Implementation of the products included in this solution	4
1.3 The roles associated with this solution	5
1.3.1 Software development roles	5
1.3.2 Product-specific roles	6
Chapter 2. Customizing the methodology and creating a lifecycle project	7
2.1 Overview of the IBM tooling and software solutions	8
2.2 Customizing the methodology and the process	8
2.3 Creating the lifecycle project	9
2.3.1 Configure the requirements management project area	13
2.3.2 Configure the change management project area	13
2.3.3 Configure the quality management project area	14
2.4 Working with requirements	14
2.4.1 Templates	15
2.4.2 Documenting the requirements	19
2.4.3 Requirements traceability	23
2.5 Creating the release and iteration plans	25
2.5.1 Creating a plan	25
2.5.2 Creating plan snapshots	27
2.5.3 Working with artifacts	28
Chapter 3. Data modeling with InfoSphere Data Architect	31
3.1 Data modeling and IBM InfoSphere Data Architect	32
3.2 Creating the IBM InfoSphere Data Architect workspace	32
3.3 Creating the data design project	34
3.4 Creating the logical data model	35
3.4.1 What a logical data model is	35
3.4.2 Creating the WARRANTY logical data model	36
3.5 Creating the physical data model	45
3.5.1 What is a physical data model	45
3.5.2 Transforming a logical data model to a physical data model	45
3.5.3 Generating the DDL	52
3.5.4 Creating the database connection	57
3.5.5 Running the DDL and creating the database	62
3.6 Importing data into the WARRANTY database	65
3.7 Exporting the WARRANTY data model to Cognos	68

Chapter 4. Decision management with WebSphere Operational Decision Management	75
4.1 Business drivers for decision management	76
4.1.1 The difference between business rules and event rules	77
4.2 Decision management with BPM	78
4.2.1 Decision management usage patterns	80
4.3 WebSphere Operational Decision Management V8	81
4.3.1 Decision Server	82
4.3.2 Decision Center	82
4.3.3 Decision Center Business Console	83
4.3.4 Decision Center Enterprise Console	83
4.3.5 Rule Designer and Event Designer	83
4.4 Overview of business rules	84
4.4.1 Building a Business Rule Project	85
4.4.2 Rule execution	90
4.5 Overview of Event Rules	92
4.5.1 Building an Event Project	93
4.5.2 Event deployment	100
4.5.3 Event testing	102
4.6 Business management of rules	105
4.6.1 Rule governance and promotion model	106
4.6.2 Using Decision Center	106
Chapter 5. Business process management with Business Process Manager	113
5.1 Overview of business process management	114
5.1.1 Managed business process	114
5.1.2 Process automation, visibility, and control	115
5.2 Overview of IBM Business Process Manager	116
5.2.1 IBM Business Process Manager configurations	117
5.2.2 Process Center	119
5.2.3 Process Server	119
5.2.4 Authoring environments	119
5.2.5 Administration tools	120
5.3 Creating the Warranty Sample	122
5.4 Add rule invocation for rules-related activities	127
5.4.1 Create a new Decision Service	127
5.4.2 Connect with WebSphere Operational Decision Management	129
5.4.3 Test the Decision Service	133
5.5 Add the user interface for human task activities	134
5.6 Enable a mediation to trigger the business process	138
5.6.1 Creating a new business object	138
5.6.2 Adding a message-based trigger to the business process	139
5.6.3 Creating a mediation flow	143
5.7 Add database integration	154
5.8 Create an advanced integration service	158
5.9 Understanding Task Types and Service Types	163
Chapter 6. Reporting with IBM Cognos Business Intelligence	167
6.1 Overview of IBM Cognos Framework Manager	168
6.1.1 Data access strategies	168
6.1.2 Metadata model	168
6.2 Creating data source connections	169
6.2.1 Configure IBM DB2 connectivity	169
6.2.2 Create the connection to the warranty claims database	169

6.3	Creating the metadata model for reporting	174
6.3.1	Create a new model and import the source metadata	175
6.3.2	Model the metadata model	180
6.3.3	Publish the model to the server	194
6.4	Authoring reports with IBM Cognos Report Studio	196
6.4.1	Creating Active Reports with Cognos Report Studio	216
6.5	Creating a dashboard with Cognos Business Insight	220
6.6	Authoring events with Cognos Event Studio	224
6.6.1	Specify an event condition	225
6.6.2	Add a task to an event condition	227
6.6.3	Manage the task execution rules	231
6.6.4	Schedule the agent	233
Chapter 7. Testing with Rational Quality Manager		235
7.1	Test planning	236
7.1.1	Creating the test plan	236
7.1.2	Creating templates	240
7.2	Test preparation	241
7.2.1	Creating test scripts	241
7.2.2	Creating test cases	246
7.2.3	Creating test suites	249
7.2.4	Storing reusable test data	250
7.2.5	Creating the test execution records	251
7.2.6	Test lab preparation	252
7.3	Test execution	253
7.4	Test traceability	257
7.5	Other useful IBM tools related to the topics in this chapter	258
Chapter 8. Deployment options with Workload Deployer		259
8.1	IBM Workload Deployer overview	260
8.1.1	Features and benefits	260
8.1.2	Cloud types	260
8.1.3	Solution elements	261
8.2	IBM Workload Deployer patterns	264
8.2.1	Cloud resources	268
8.3	Administrative interfaces	270
8.3.1	Web-based user interface	270
8.3.2	Command-line interface	271
8.3.3	Representational State Transfer API	272
8.4	Appliance settings	272
8.5	Tools for building custom assets	273
8.5.1	IBM Image Construction and Composition Tool	274
8.5.2	Plug-in Development Kit	277
8.6	Deploying virtual patterns to the cloud	278
8.6.1	Deployment planning	279
8.7	Applying a virtual application pattern to our scenario	280
8.8	Additional information	282
Appendix A. Additional material		285
	Locating the web material	285
	Restoring the IBM Cognos Business Intelligence examples	285
	Restoring the Framework Manager model	285
	Restoring the IBM Cognos Business Intelligence deployment archive	286

Appendix B. Creating a messaging infrastructure and web user interface.	287
Creating the messaging infrastructure	287
Defining a service integration bus queue	287
Defining JMS resources	289
Web user interface configuration	293
Creating a dynamic web project	293
Creating a servlet	294
Creating an HTML page	296
Configuring the servlet to point to the JMS components.	297
Running the web user interface application	298
Related publications	299
IBM Redbooks	299
Help from IBM	299

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 grant 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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


COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.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®	IBM SmartCloud™	PureSystems™
AppScan®	IBM®	Rational Team Concert™
CICS®	ILOG®	Rational®
Cognos®	IMS™	Redbooks®
DataPower®	Informix®	Redbooks (logo)  ®
DataStage®	InfoSphere®	Rhapsody®
DB2 Universal Database™	Jazz™	System z®
DB2®	OmniFind®	Tivoli®
developerWorks®	PowerVM®	WebSphere®
Global Business Services®	PureApplication™	z/VM®

The following terms are trademarks of other companies:

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

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Preface

In this IBM® Redbooks® publication we describe how to build an advanced business application from end to end. We use a fictional scenario to define the application, document the deployment methodology, and confirm the roles needed to support its development and deployment. Through step-by-step instructions you learn how to:

- ▶ Define the project lifecycle using IBM Solution for Collaborative Lifecycle Management
- ▶ Build a logical and physical data model in IBM InfoSphere® Data Architect
- ▶ Confirm business rules and business events using IBM WebSphere® Operational Decision Management
- ▶ Map a business process and mediation using IBM Business Process Manager
- ▶ Use IBM Cognos® Business Intelligence to develop business insight

In addition, we articulate a testing strategy using IBM Rational® Quality Manager and deployment options using IBM Workload Deployer.

Taken together, this book provides comprehensive guidance for building and testing a solution using core IBM Rational, Information Management, WebSphere, Cognos and Business Process Management software. It seeks to demystify the notion that developing and deploying advanced solutions is taxing.

This book will appeal to IT architects and specialists who seek straightforward guidance on how to build comprehensive solutions. They will be able to adapt these materials to kick-start their own end-to-end projects.

The team who wrote this book

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

Ahmed Abdel-Gayed is an IT Specialist at the Cairo Technology Development Center (CTDC) of IBM Egypt since 2006. He has worked on a business intelligence application that provides business reporting for IBM Support personnel and IBM managers worldwide. His role was to author and support Cognos dashboards, reports, and the business model. He has also worked on a variety of search-related projects to implement IBM OmniFind® Enterprise Search solutions. He also has experience in application development with Java, J2EE, and IBM DB2®. Ahmed holds a bachelor's degree in Computer Engineering from Ain Shams University, Egypt.

Kulvir Singh Bhogal is a Product Manager in the United States. He has 16 years of experience in the software engineering field. He has worked at IBM for 12 years. His areas of expertise include WebSphere, Java, and cloud-based technologies. He has written numerous articles and contributed to books about IBM software.

Don Carr is an Operations Specialist in IBM Software Group, Research Triangle Park. He has 15 years of experience in software and services billing, enablement, support and development. His current focus as a member of the SOA Sandbox team is making IBM software more accessible for trial and evaluation by both customers and partners. He earned an MBA from Florida Atlantic University.

Richard Davies is a WebSphere Operational Decision Management Technical Account Manager. Richard is based in Melbourne, Australia.

Aditya P Dutta is an IT Architect at IBM Global Business Services®, India. He provides consulting and architectural assistance to IBM clients for large and complex initiatives. He specializes in BPM, SOA, and Enterprise Integration solutions using IBM middleware products. Aditya holds a Bachelor's degree in Electronics and Instrumentation Engineering from the College of Engineering and Technology, Bhubaneswar, India. He has published and presented at various IBM technical forums including developerWorks® and technical conferences, and has co-authored IBM Redbooks publications.

Marcelo Correia Lima is a Business Intelligence Architect at IBM. He has 15 years of experience in leading, developing, and integrating Enterprise Applications. His current main area of expertise is Business Analytics Optimization (BAO) Solutions. He has been planning and managing full life cycle implementation of many projects, involving Multidimensional Modelling, Multidimensional Clustering and Partitioning, IBM InfoSphere Data Architect, IBM DB2, IBM InfoSphere DataStage® and IBM Cognos Business Intelligence. Before working as Business Intelligence Architect, he worked on design and implementation of IBM WebSphere and Java Enterprise Edition Applications for IBM Data Preparation/Data Services.

Agueda Martinez Hernandez Magro is a Software Development Manager in IBM Rational Application Developer. She is located at the Mexico Software Lab, in Guadalajara. She has 14 years of experience in the software development field. She holds a Master Degree in Computer Science from the Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) Campus Guadalajara. Her main area of expertise is the software development process.

Yuka Musashi is a remote technical support representative of WebSphere products. She is located in Japan and worked as a specialist in Application and Integration Middleware. She meets clients' expectations by analyzing and providing the solution for their problems. She graduated from AoyamaGakuin University as a bachelor.

Michael Norris is a Senior Product Manager for IBM Cognos Business Intelligence and Platform working in Ottawa, Ontario, Canada. With over 12 years experience in business intelligence and information management, Michael has focused on meshing software solutions with the technology and infrastructure that organizations use to drive business results and outperform their competition. Michael studied Computer Engineering Technology at St. Lawrence College in Kingston, Ontario, Canada and currently focuses on cloud solutions and enablement technologies for Business Analytics.

Felix Pistorius is a corporate student in Germany. He has been working two years for IBM and is aiming for a B.Sc in Applied Computer Science from University of Corporate Education, Mannheim. Felix participated on this book while on an internship.

This project was led by:

Martin Keen is a Redbooks Project Leader in the Raleigh Center. He primarily leads projects about WebSphere products and service-oriented architecture (SOA). Before joining the ITSO, Martin worked in the EMEA WebSphere Lab Services team in Hursley, U.K. He holds a Bachelor's degree in Computer Studies from Southampton Institute of Higher Education.

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

- ▶ Torbjorn Johansson, IBM Business Process Optimization
- ▶ Steve Demuth, IBM Senior Technical Staff Member, WebSphere Decision Management
- ▶ Robert D. Johnson, Director, Advanced Platform Development

- ▶ Raphael Chancey, IBM Senior Technical Staff Member, Chief Architect, Smarter Platform
- ▶ Salil Ahuja, IBM Industry Accelerators, Business Process Optimization
- ▶ Debbie Willmschen, IBM International Technical Support Organization, Raleigh Center
- ▶ Linda Robinson, IBM International Technical Support Organization, Raleigh Center
- ▶ Alfred Schwab, editor, IBM International Technical Support Organization, Poughkeepsie Center
- ▶ Barbara McKee, STSM, Advanced Platform Development, IBM Application and Integration Middleware Software
- ▶ Greg Flurry, IBM Distinguished Engineer, Business Process Optimization
- ▶ Chris McPherson, IBM Product Manager, Cognos Framework Manager and Metadata
- ▶ Jeff MacDonald, IBM Product Manager, Cognos Active Reports
- ▶ Dean Browne, IBM Product Manager, Cognos Event Studio
- ▶ Marcus Eugenio Facanha Braga, IBM BAO Delivery Manager
- ▶ Henrique Messa da Silva, IBM Business intelligence Application Developer
- ▶ Andreia Stelet, IBM IT Specialist, Global Business Services
- ▶ Joe Wigglesworth, IBM Senior Technical Staff Member, PureApplication™ System Development
- ▶ Matthew G. Kelm, IBM Software Engineer, IBM Software Group
- ▶ Dustin Amrhein, IBM WebSphere Client Technical Professional
- ▶ Nick Norris, Solution Architect "UnLeash the Labs", IBM Rational
- ▶ Dave Tropeano, Senior Manager, Solution Architecture, IBM Rational
- ▶ Isabel Torres, FVT Leader, Rational Requirements Composer, IBM Rational
- ▶ Sergio Jose Deras Arreola, Rational Team Concert™ Install Development Leader, IBM Rational

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email 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

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>



Introduction to the solution and the scenario used in this book

In today's fast-paced business climate, companies must make quick decisions in order to win new business, maintain existing relationships and build on the trust and goodwill earned over time. However, due to the complexity of existing business logic, adjusting or improving applications takes careful planning and time to implement any needed changes effectively. This IBM Redbooks publication documents the development and deployment of an integrated sense-and-respond solution that provides awareness through events, direction through decisions, actions through processes, and insight through analysis. This simplified solution will demonstrate how you can structure advanced applications around business process management concepts and methodologies to provide decision-making capabilities based on real-time data analysis.

This chapter describes the representative business scenario that will be used throughout this book in the following sections:

- ▶ Business scenario summary
- ▶ The solution architecture used in this book
- ▶ The roles associated with this solution

1.1 Business scenario summary

The business scenario in this book involves a warranty reporting solution that accepts post-purchase claims for product repair or replacement from the original purchaser (that is, customer), that processes all claims via an online reporting tool. This tool is the front end to a software solution that uses information provided by the customer via the warranty claim to determine the appropriate course of action the business should take (that is, refund, repair, or replacement) and to track and report claim trends. This analysis enables product teams and managers to understand common problems in order to correct or improve the problematic feature or functions in future releases.

Our sample scenario consists of the following steps:

1. A customer provides the serial number of a purchased product and chooses the claim type via the online reporting tool.
2. While the tool logs each claim, if it records multiple claims for the same serial number in quick succession, it triggers an event indicating that these claims need to be examined together.
3. Upon receipt of each claim, the event manager launches a business process to initiate the warranty review.
4. This process invokes the business rules system to assess each event and invoke the appropriate actions per the documented warranty policies (that is, a claim can be processed automatically or similar claims require special handling).
5. This process also writes the claim status to an attached database.
6. The information gathered during the warranty review process gets consolidated in reports via the decision support system, which is accessible by web browser or mobile device.

This scenario demonstrates how business leaders can leverage events through well-defined processes to create awareness, make decisions, and act accordingly in order to become more flexible in facing rapidly changing business situations. Common problems can be addressed automatically, while unique concerns can be explored and resolved by decision makers who are enabled to act tactically and strategically via integrated reporting and analysis.

The benefit to your customers is a more responsive system. While our sample solution is not production ready, it does articulate how all the necessary components fit together and can be used effectively to build such a system. It lays the foundation for you to become proactive, which should help you win new business, maintain existing relationships, and build on the trust and goodwill you have earned through the years.

1.2 The solution architecture used in this book

This section summarizes how IBM Business Process Manager, IBM WebSphere Operational Decision Manager, and IBM Cognos Business Intelligence can be integrated to create situational awareness, action, and insight. It also highlights how Rational tooling can be used to manage the complexities associated with developing and deploying advanced applications. The IBM Solution for Collaborative Lifecycle Management (CLM) provides integrations across Jazz™-based products to connect the work of analysts with development and test teams, which enables them to validate solution requirements and produce solution artifacts more efficiently.

Figure 1-1 illustrates the architecture for the warranty reporting solution.

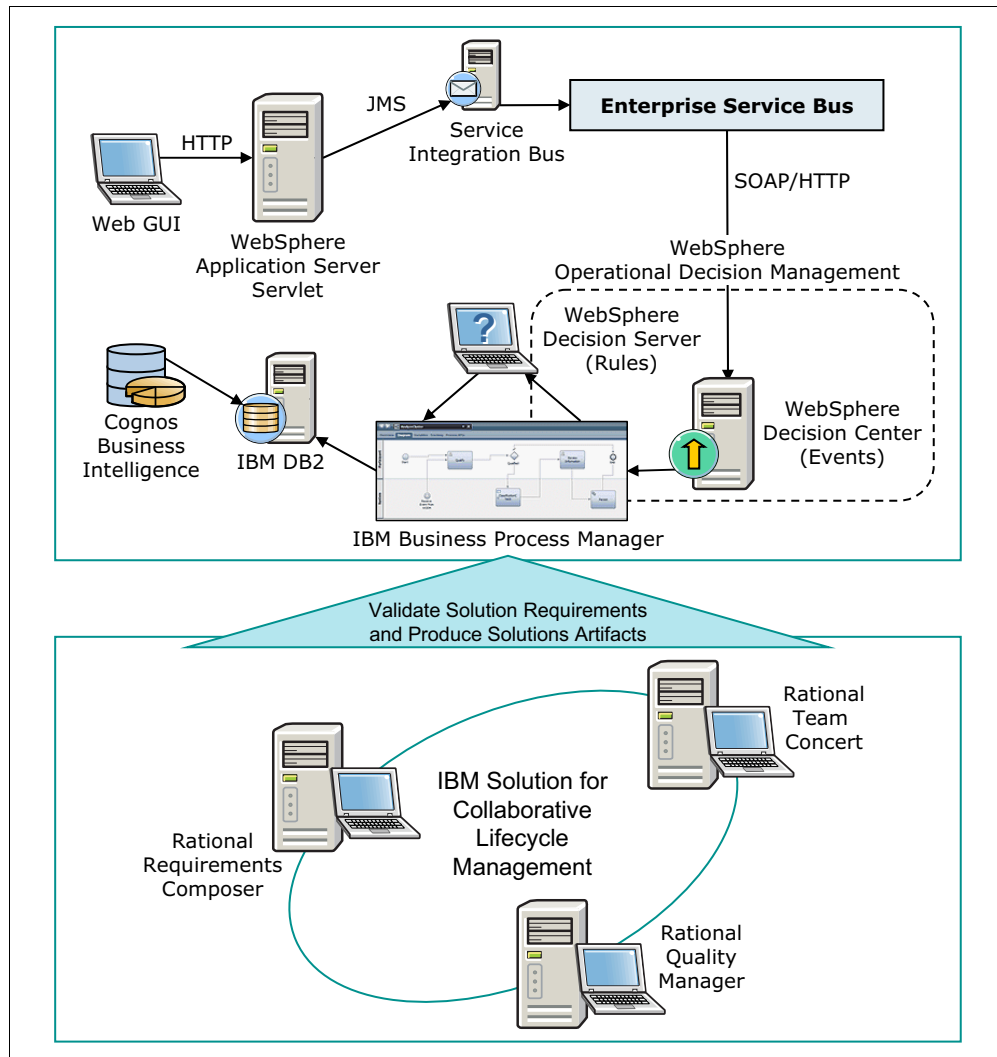


Figure 1-1 Warranty reporting solution architecture

1.2.1 IBM products used in the solution

Table 1-1 lists the IBM products that are used in our sample solution. The justification for selecting each product is discussed in 1.2.2, “Implementation of the products included in this solution” on page 4.

Table 1-1 List of IBM products used in this solution

Name	Function within this solution
IBM Business Process Manager Advanced V8	Provides a comprehensive BPM platform. The Advanced edition supports high-volume automation and extensive system integration as well as human workflow. The main components of this product are the Process Server, Process Center, IBM Process Designer, and IBM Integration Designer.

Name	Function within this solution
IBM Cognos Business Intelligence V10.1.1	Provides mechanisms for querying, reporting, and analyzing data. Provides dashboards and the software to gather and organize information from multiple sources to help make decisions for better business outcomes now and in the future.
IBM Infosphere Data Architect V8.1.0.0	Provides a collaborative data design environment for discovering, modeling, relating, standardizing, and integrating diverse and distributed data assets.
IBM WebSphere Operational Decision Management V8	Provides a comprehensive platform for the management and execution of business rules and business events: WebSphere Decision Center enables business users to govern business rules and business event-based decision logic. WebSphere Decision Server automates decision logic, enabling sense and respond actions based on context of an event.
IBM Solution for Collaborative Lifecycle Management V4.0	Provides seamless integration that connects the work of the analyst, the developer, and the test team with bundled IBM Rational Team Concert V4.0, IBM Rational Requirements Composer V4.0, and IBM Rational Quality Manager V4.0.

1.2.2 Implementation of the products included in this solution

This solution requires the implementation of the following products:

- ▶ IBM Solution for Collaborative Lifecycle Management (CLM)

The tooling included in this solution was used to manage the build, testing, change and configuration management of the warranty reporting solution. Tooling usage is highlighted in Chapter 2, “Customizing the methodology and creating a lifecycle project” on page 7 and Chapter 7, “Testing with Rational Quality Manager” on page 235.
- ▶ IBM Infosphere Data Architect

This tool was used to organize and manage both the logical and physical databases required by our sample solution. Tool usage is highlighted in Chapter 3, “Data modeling with InfoSphere Data Architect” on page 31.
- ▶ IBM Business Process Manager

IBM Business Process Manager was used to implement the business process for our scenario. When business events are passed by IBM WebSphere Operational Decision Manager, business decisions are invoked to satisfy the process. Tool usage is highlighted in Chapter 5, “Business process management with Business Process Manager” on page 113.
- ▶ IBM WebSphere Operational Decision Manager

This product was used to define necessary actions to events triggered by the warranty reporting process. It enables us to automate claims decisions based on specified business situations defined in the business process. Tool usage is highlighted in Chapter 4, “Decision management with WebSphere Operational Decision Management” on page 75.
- ▶ IBM Cognos Business Intelligence

This tooling was used to provide insight through the creation of reports gathered from the data and events collected by the warranty reporting system. Tooling usage is highlighted in Chapter 6, “Reporting with IBM Cognos Business Intelligence” on page 167.

1.3 The roles associated with this solution

Highlighting the roles needed to support the design, development and implementation of our sample will allow you to understand the required investment in skills that will be required to duplicate this effort. Table 1-2 lists the roles that are associated with this solution.

Table 1-2 Software development and product-specific roles

Software development roles	Product	Product-specific roles
Analyst	IBM Business Process Manager	Process analyst
	IBM WebSphere Operational Decision Manager	Business analyst
Data developer	IBM Infosphere Data Architect	Data architect
	IBM Cognos Business Intelligence	Cognos BI Administrator
Business developer	IBM Business Process Manager	<ul style="list-style-type: none"> ▶ Process participant (SME) ▶ Process developer ▶ Integration developer ▶ Software administrator ▶ Solution architect
	IBM WebSphere Operational Decision Manager	<ul style="list-style-type: none"> ▶ Policy manager ▶ Rule and event author ▶ Software administrator ▶ Solution architect
Test engineer		
Project leader		

1.3.1 Software development roles

Software development roles define the level of effort required from the associated product-specific roles. These main roles are centered around standard project management conventions. The software development roles used in this book are:

- ▶ Analyst - Clarifies customer requirements and manages them using IBM Rational Requirements Composer.
- ▶ Data developer - Creates the data model and its infrastructure based on solution requirements and then implements both.
- ▶ Business developer - Creates the business processes, decisions, situations, analysis reports and dashboards according to approved business requirements, and designs the integration between the process and business rules, the database, and other components.
- ▶ Test engineer - Describes the test case for both the unit test with IBM Rational Quality Manager and IBM Rational Team Concert and the integration test with IBM Rational Quality Manager, and evaluates test cases.
- ▶ Project leader - Organizes and leads the project and communicates with the customer.

1.3.2 Product-specific roles

Product-specific roles are detailed roles that are required for each product used in the development of the solution. If you were implementing this solution, you would want to plan on having one or more of these roles performed by suitably qualified staff:

- ▶ IBM Infosphere Data Architecture
 - Data architect - Responsible for the data architecture; leads the data modeling to create the DDL and the data design for the database.
- ▶ IBM Cognos Business Intelligence
 - Cognos BI Administrator - Responsible for administering server environments, creating and maintaining metadata, as well as generating all types of fixed reporting and ad hoc dashboards to meet the needs of the organization.
- ▶ IBM Business Process Manager
 - Process analyst - Analyzes the process from a business perspective and leads process improvement once the as-is process is implemented.
 - Process participant (SME) - Provides additional depth and detail on process flows, business policies, and user interface interactions as discovered in user stories.
 - Process developer - Creates and implements process flows, services, business logic, and user interfaces for human workflow tasks (known as coaches in IBM Business Process Manager).
 - Integration developer - Organizes the integration of business processes and business rules as well as other back-end systems such as databases, queues or ERPs. This role is also responsible for the implementation of the Enterprise Service Bus.
 - Software administrator - Designs and sets parameters, such as custom data storage with the administrative console, and deploys the process application to the run time.
 - Solution architect - Designs the overall solution architecture of the process application including how interaction between the process layer, the integration layer and the rules and events should function.
- ▶ IBM WebSphere Operational Decision Manager
 - Business analyst - Designs and writes the specifications for business and event rules and validates the executions of each.
 - Policy manager - Analyzes business and event rule execution and maintains each.
 - Rule and Event author - Creates and updates business and event rules with the designer tool.
 - Software administrator - Designs, implements, and manages the administrative matters of WebSphere Operational Decision Manager.
 - Solution architect - Designs the overall solution architecture of the business rules application and leads this staff.



Customizing the methodology and creating a lifecycle project

When organizations start to implement Business Process Management, they can experience delays due to differing methodologies employed by their business analysts and their development or test teams. Any differences can impact effective communication and timely delivery, because these challenges will likely cause rework to ensure that all developed assets meet defined expectations.

Having a common methodology, established and documented processes, and collaborative tools that team members can use to understand business expectations effectively and communicate directly are the keys needed to attain agility. In this chapter we highlight the tooling and software solutions available from IBM that are designed to help you maximize collaboration between your development and test teams.

IBM Solution for Collaborative Lifecycle Management combines Rational Team Concert, Rational Quality Manager, Rational Requirements Composer into one image that provides an integrated platform to coordinate software development activities across business and system requirements, design, development, build, test, and delivery.

We also discuss the requirement for customizing IBM Solution for Collaborative Lifecycle Management, which will guide the development process for implementing our warranty claims sample (that is, links business process management with operational decision management).

2.1 Overview of the IBM tooling and software solutions

We discuss the following tools:

The IBM Solution for Collaborative Lifecycle Management is a set of seamlessly integrated tools that work together as one: IBM Rational Requirements Composer, IBM Rational Team Concert, and IBM Rational Quality Manager.

- ▶ IBM Rational Requirements Composer provides a collaborative way to define and manage requirements.
- ▶ IBM Rational Team Concert is a collaborative environment that integrates agile planning, tasks tracking, source control, continuous builds and a configurable development process.
- ▶ IBM Rational Quality Manager is a collaborative test environment to plan, develop, execute, monitor and report testing.

The interaction between these tools is transparent due to the associations between artifacts defined on each product. Figure 2-1 shows the typical interaction.

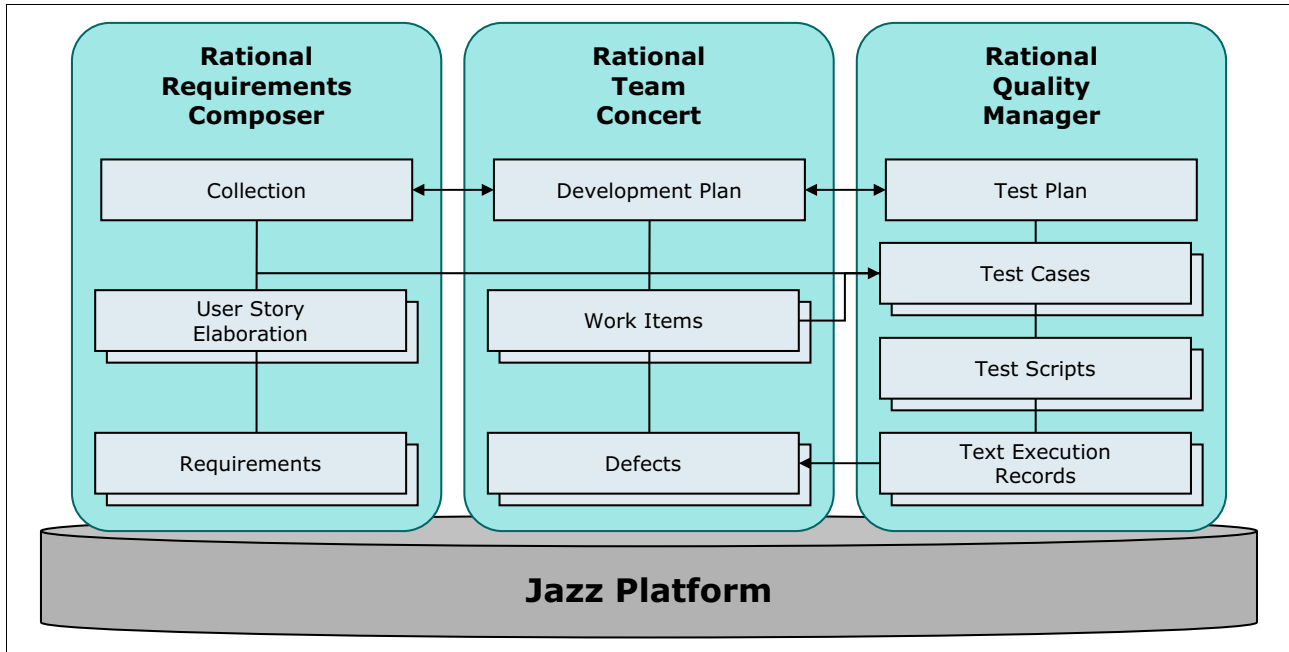


Figure 2-1 Collaborative Lifecycle Management

Version 4.0 of the IBM Solution for Collaborative Lifecycle Management is enhanced with a beta version IBM Rational Software Architect Design Manager and IBM Rational Rhapsody® Design Manager. These new tools provide decision management capabilities to architect, design and develop software and systems. To learn more about these new tools, click the following link to access product information on Jazz.net:

<https://jazz.net>

2.2 Customizing the methodology and the process

Documenting the methodology and processes is a good way to start any project. Methodologies and processes help companies to achieve results in an organized and

repeatable way. When tools and processes are not integrated, people tend to feel overwhelmed, including your development team.

Companies leveraging business process management principles can benefit from the use of proven methodologies such as the ILOG® Solution Implementation Standard (ISIS), which consists of product-specific best practices and artifacts that maximize effectiveness because they were created from multiple project engagements.

ISIS is built on the Unified Method Framework (UMF) and Open Unified Process (OpenUP), which is a well-defined lifecycle software development process and industry standard that provides a disciplined approach for task and responsibility management. Over the years, it has been used successfully in Business Process Management projects throughout the world.

IBM Solution for Collaborative Lifecycle Management tools allow you to use OpenUP on Business Process Management projects such as our warranty claims sample.

ISIS: To learn more about ISIS, we refer you to:

Combining Business Process Management and Enterprise Architecture for Better Business Outcomes, SG24-7947

OpenUp: To learn more about OpenUp, we refer you to:

<http://epf.eclipse.org/wikis/openup/>

2.3 Creating the lifecycle project

You need to create the lifecycle project to guide your team through the implementation phase.

A lifecycle project will tie together all the elements used in a software development project. With a lifecycle project you are able to do real-time planning from requirements, development, testing, and even defect management. In addition, you gain the ability to perform lifecycle traceability.

To get more information about the benefits of a lifecycle project, go to:

<https://jazz.net/products/clm/>

In the IBM Solution for Collaborative Lifecycle Management V4, you will create the lifecycle project and the containers needed for your project artifacts using Rational Requirements Composer, Rational Team Concert, and Rational Quality Manager.

Note: The following instructions assume the Jazz Team Server is up and running. For more information on installation steps go to:

https://jazz.net/help-dev/clm/topic/com.ibm.jazz.doc/topics/c_planning_install.html

To create a lifecycle project:

1. In the Server Administrator page, click the **Create Lifecycle Project** link under the Manage Lifecycle Projects section as shown in Figure 2-2 on page 10.


Manage Lifecycle Projects

Why Create and Manage Lifecycle Projects? Using Lifecycle Projects allows you to easily establish and manage associations between projects in the software development lifecycle. Whether you plan to make associations when you first create a project or later, starting with a Lifecycle Project gives you more flexibility to easily make these associations than if you were to create a single standalone Project Area within an application.

-  [Create Lifecycle Project](#)
-  [Manage Lifecycle Projects](#)
-  [Try out the Money That Matters sample](#)
-  [About creating and managing lifecycle projects](#)

Figure 2-2 Creating the lifecycle project

2. Set the project name, for instance BPM_ODM.
3. Now select a base process by selecting **Agile Requirements, OpenUP Process** which is the base of the ISIS methodology, and **Quality Manager Default Process**, as shown in Figure 2-3.

 **BPM_ODM** *

Template:
This template sets up associated project areas for quality management, requirements management, and change and configuration management. The testers (in quality management) and analysts (in requirements... Show More...

Locale: ▼

Description:

Artifact Containers






Artifact Containers for Template (3)		
Method	Artifact Containers	Process / Project Template
Create:	<input type="text" value="BPM_ODM (Requirements) 'Requirements Management'"/> <small>↳ Provides  Requirements to BPM_ODM (Quality Management)</small>	project area in '/rm' for <input type="text" value="Agile Requirements Templ:"/> ▼
Create:	<input type="text" value="BPM_ODM (Change Management) and Configuration Management"/> <small>↳ Provides  Implementation Requests to BPM_ODM (Requirements) ↳ Provides  Defects to BPM_ODM (Quality Management) ↳ Provides  Quality Management Tasks to BPM_ODM (Quality Management) ↳ Provides  Requirements Change Requests to BPM_ODM (Requirements)</small>	project area in '/ccm' for 'Change and Configuration Management' <input type="text" value="OpenUP Process"/> ▼
Create:	<input type="text" value="BPM_ODM (Quality Management) Management"/>	project area in '/qm' for 'Quality Management' <input type="text" value="Quality Management Defau"/> ▼

Figure 2-3 Processes for the lifecycle project

4. After clicking **Save**, the project areas are created and the applications are ready to be used.
5. If you did not have users previously created, create them now by going to the Jazz Team Server home, then navigate to **Users** → **Create User**.

6. Provide the user name, name and email and select the appropriate client access licenses for the user, and repeat this step for all the users that you need.

Note: A project area is the representation in the system of a project. The project area contains the artifacts and the relationships between them.

The lifecycle project has been created. To navigate across the project areas you use URLs in the form of:

https://server:port/jts	for the Jazz Team Server
https://server:port/rm/web/console/BPM_ODM	for requirements
https://server:port/ccm/web/console/BPM_ODM	for change and configuration management
https://server:port/qm/web/console/BPM_ODM	for quality manager

You will want to familiarize yourself with the different menus used in the server and project areas. The following figures highlight the various options for each project artifact (Figure 2-4, Figure 2-5, Figure 2-6, and Figure 2-7).

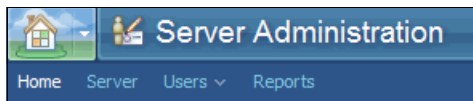


Figure 2-4 Server Administration menu

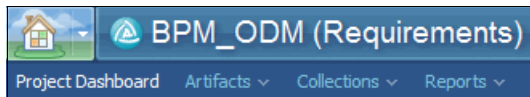


Figure 2-5 Requirements menu

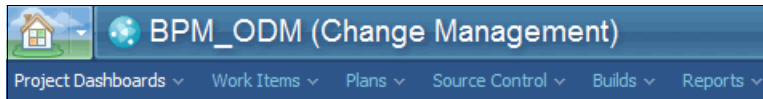


Figure 2-6 Change management menu

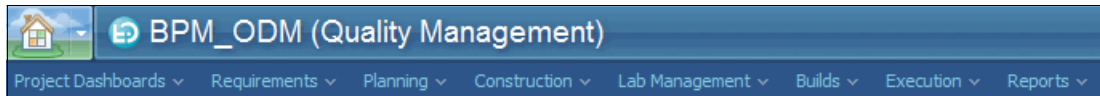


Figure 2-7 Quality management menu

In addition to these menus, there are also the User Profile, Administration, and Help menus to provide access to software-oriented tasks. It is important to note that the options listed under the Administration menu will vary between applications. There is also a very useful Quick Search menu that enables you to find artifacts quickly. See Figure 2-8, Figure 2-9, and Figure 2-10 on page 12, and Figure 2-11 on page 13.

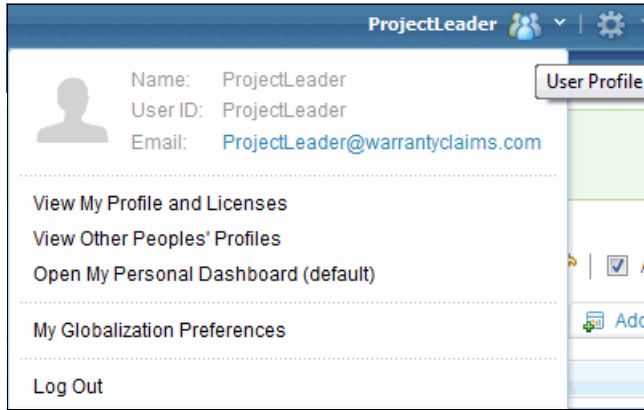


Figure 2-8 User profile menu

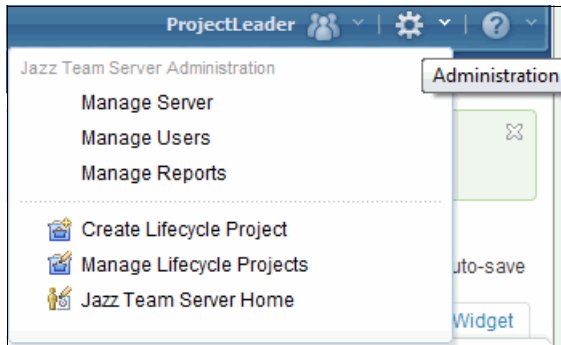


Figure 2-9 Administration menu

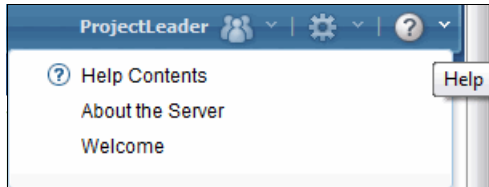


Figure 2-10 Help menu

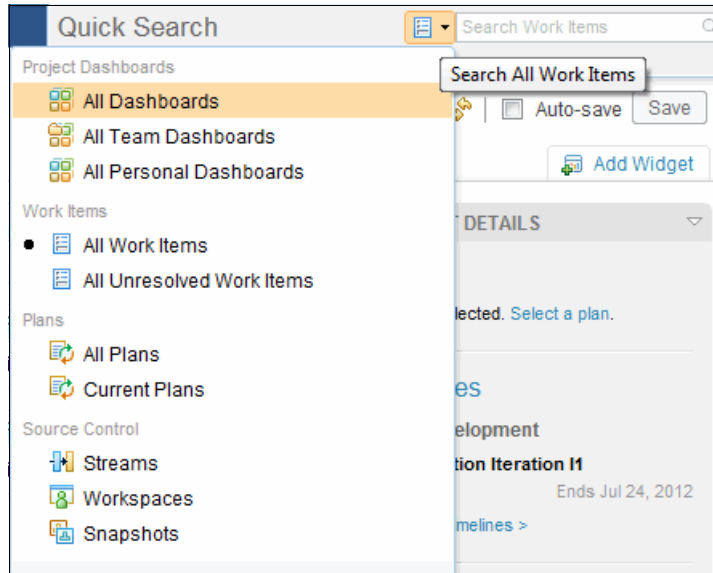


Figure 2-11 Quick search menu

Now it is time to configure each project area or the containers for the various artifacts we need to work in this development project.

2.3.1 Configure the requirements management project area

The requirements project area contains artifacts that analysts create to guide the development and test teams during the implementation phase.

You configure the requirements project area as follows:

1. Navigate to the Jazz Server Administration and go to **Home**.
2. Look for Manage Lifecycle Projects and select the recently created project in our example BPM_ODM.
3. Go to the requirements project area by clicking the appropriate link, which is project dependent.
4. To add the users that will have access to your project, click the **Add...** link in the Members section.
5. Next select the **Process Roles** in each row, or for each user.

Note: Make sure the Business Process Management and Operational Decision Management Rule Analysts get an author role so they can create artifacts. Other users just need to be able to add comments to existing requirements, but not to add new or modify existing ones. This will protect the integrity of the requirement artifacts.

6. Add a project area administrator that is able to modify the project area configuration.
7. To make other changes, navigate to the specific options menu and follow the prompts.

2.3.2 Configure the change management project area

The change management project area contains the planning and development artifacts you create in a development project.

To configure the change management project area:

1. Add the users that will have access to your project area by clicking the **Add...** link in the Members section and associate the process roles.
2. Add a project administrator.
3. On the right side locate the Team Area Hierarchy, click **Create Team** to create a team area. Traditionally, you can create a team area for each component team that works in your project; the creation of it is very similar to the project area.

Team area: A team area manages the team artifacts, roles and membership. You can create a team area for each component team and restrict the access to just the required members to protect the team's assets.

4. Navigate to Timelines in the left menu. Since you are using a predefined template, a main development project timeline, a release and some iterations are already created.
5. You can customize the roles, permissions, iteration types, access control and process definition as needed.
6. Go to Releases in the same menu and click the green plus symbol to add the releases for which you want the work items to be created against.
7. Similar to the releases setup, you need to create categories so that you can assign the work items to a specific team. In categories you can create hierarchies to increase granularity.
8. If you navigate to Work items in the left menu, you can see the predefined types of work items and which fields they contain. You can modify the existing attributes or even add new types of work items according to your project needs.

2.3.3 Configure the quality management project area

Configuring the quality management project area is very similar to the configuration of the change management project area. In Rational Quality Manager you can create quality associated tasks and assign these to team areas.

A special configuration that is particular to Rational Quality Manager is the setup of the test artifacts workflow, which indicates, as per the name, the flow of the different status an artifact can have. The default configuration is appropriate for our project.

2.4 Working with requirements

One of the common practices that successful software projects share is that they have clearly documented, unambiguous requirements. Traditionally, requirements are documented in static artifacts that soon become obsolete due to the frequency of change in modern business that affects requirements. Time is always a constraint, so having to rework the requirements document is hard and often neglected. The problem is compounded when, after undocumented changes are made, the system ends up doing things that the users do not expect.

Proper requirements management helps teams to unify their view about user expectations and needs. It also reduces rework and increases quality.

2.4.1 Templates

Business process management and operational decision projects use a particular business language. In this kind of projects you have specific needs and you may need specific artifacts to document them. Before you start creating your requirements you will want to customize your project and artifacts.

To customize your requirements project, in Rational Requirements Composer, go to **Administration** → **Manage project properties**, as shown in Figure 2-12.

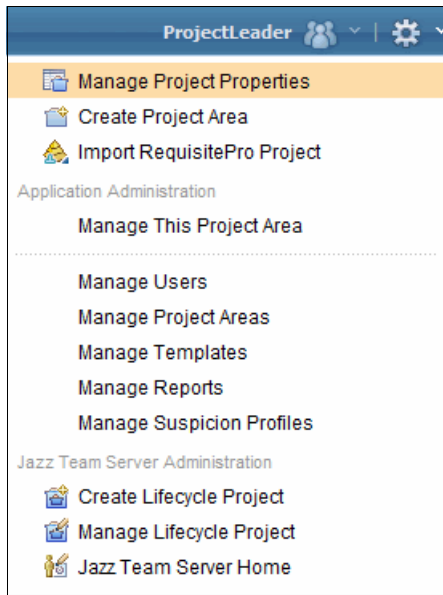


Figure 2-12 *Manage project properties*

Depending on your needs, navigate to the appropriate tabs to make the needed adjustments. For example, you can create new artifact types, new attributes, data types, or links between artifacts. See Figure 2-13 on page 16.

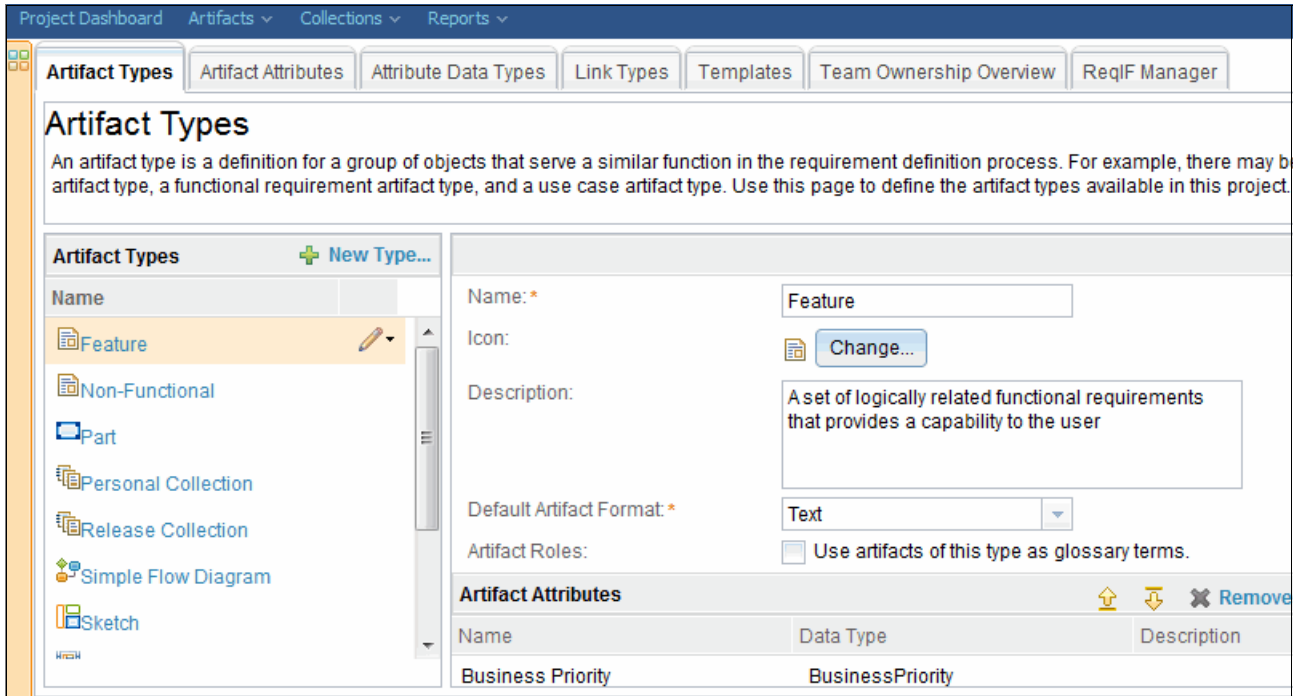


Figure 2-13 Customizing the project

After creating all your artifacts, you can create a project template so that you can reuse all your new artifacts; Figure 2-14.

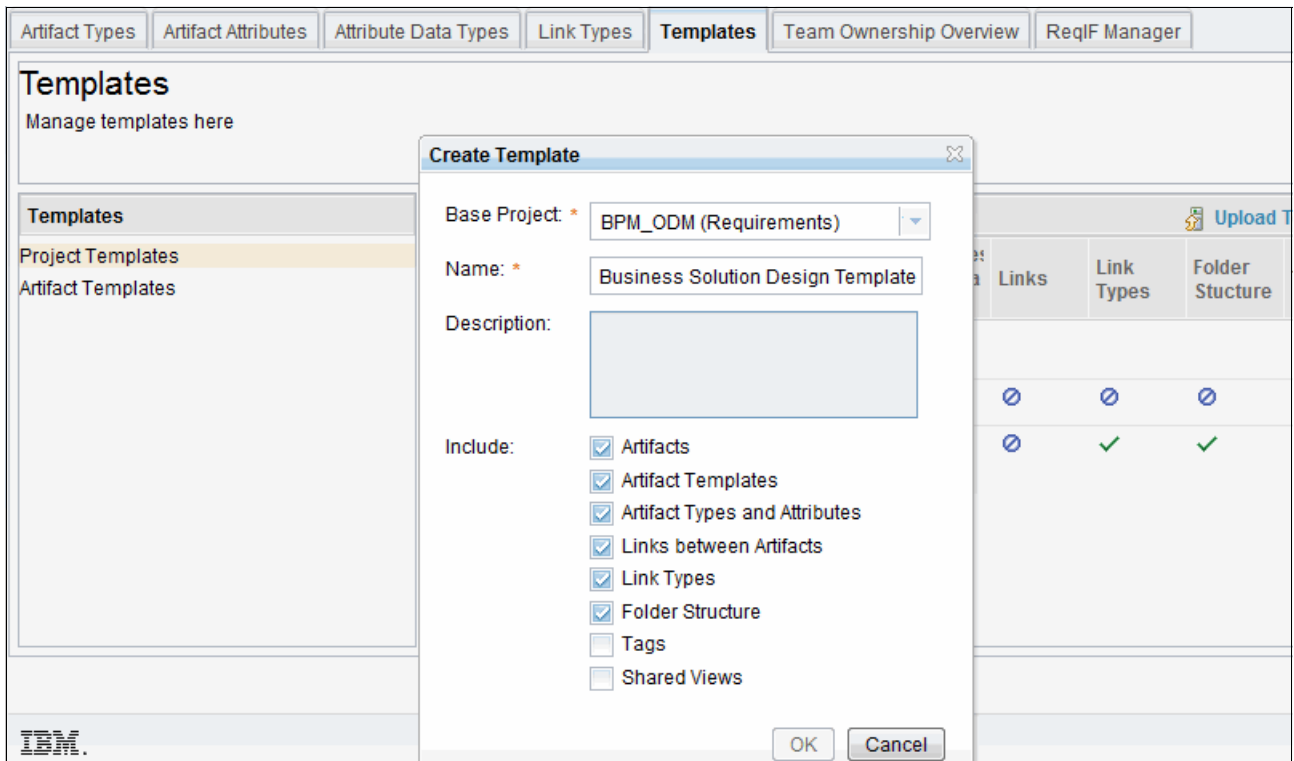


Figure 2-14 Project template

To create a new project, you can go to **Administration** → **Create Project Area**. You will be prompted to enter a name, for instance NewProjectArea, and select a project template. In this example we named the project Business Solution Design Template, as shown in Figure 2-15.

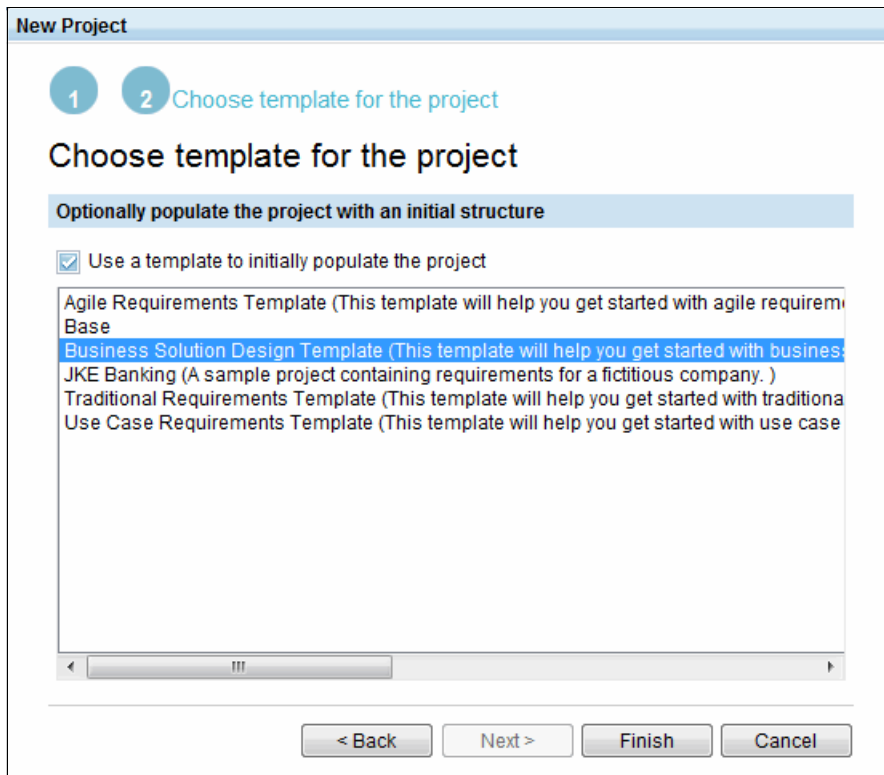


Figure 2-15 Create a project from a template

Your new project is created with the artifacts you defined, which are now part of your project template. See Figure 2-16 on page 18.

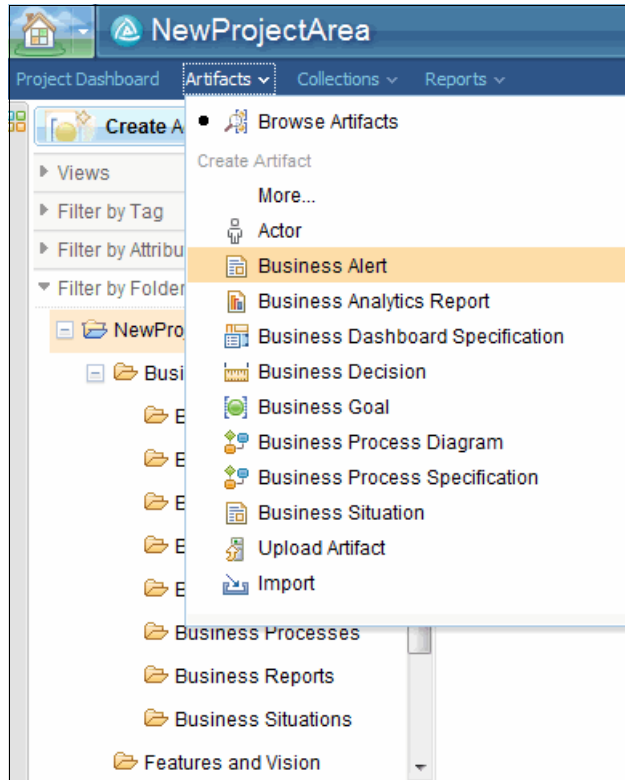


Figure 2-16 New project area

When you create new artifacts, you need to select the artifact template you created so that it populates the content you defined in the template. See Figure 2-17.

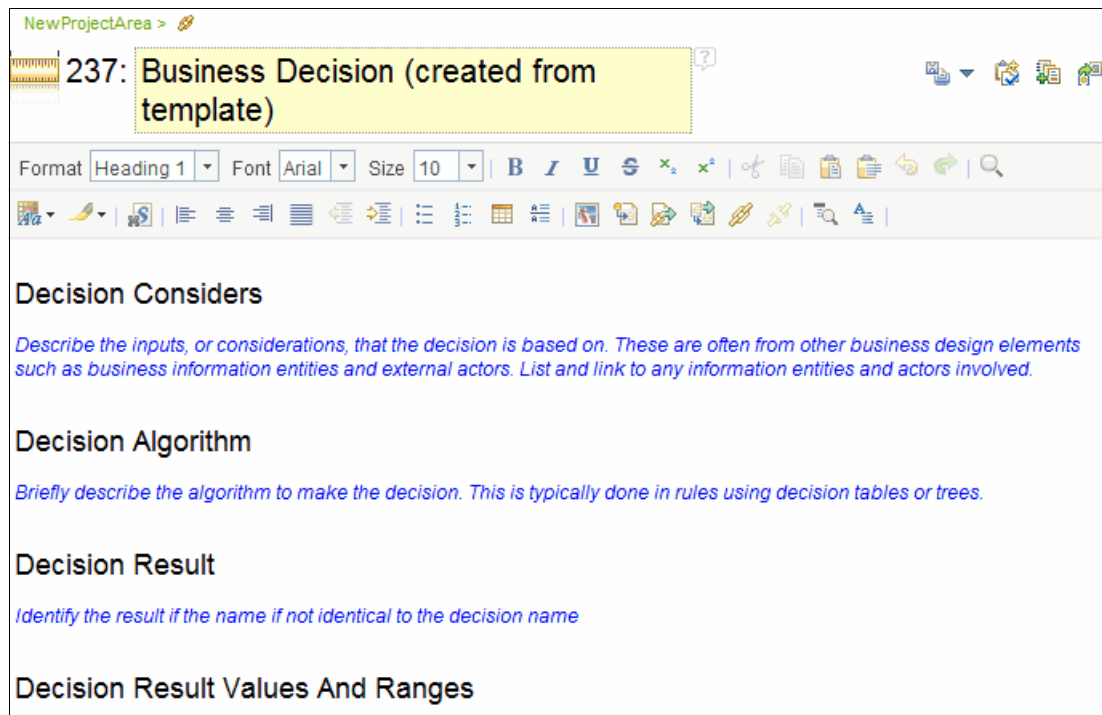


Figure 2-17 Creating an artifact from a template

2.4.2 Documenting the requirements

Now the requirements project area is ready for the team to use. Depending on the process or project template selected when creating the lifecycle project or a customized project area for requirements management, different artifacts can be used to document the requirements.

To create artifacts, navigate to Artifacts and select the type of artifact you need, as shown in Figure 2-18, which shows the artifacts defined in the Agile Requirements template.

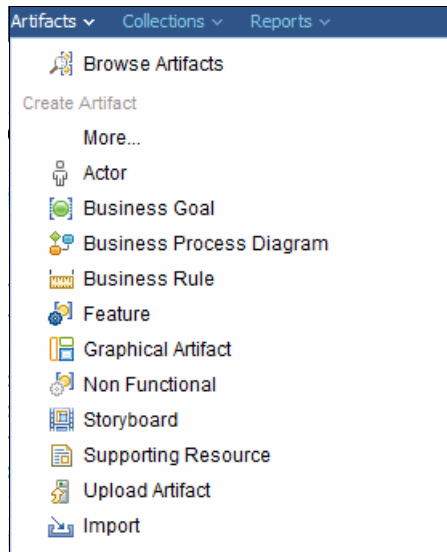


Figure 2-18 Creating requirements artifacts

During the collection of information, analysts can use artifacts such as storyboards, user interface sketches, simple flow diagrams or terms, or your customized artifacts. Analysts can also upload existing artifacts or other support resources for use via the tool.

All these artifacts can be created via Rational Requirements Composer, which provides the necessary tooling in a user-friendly environment that is well documented. Figure 2-19 on page 20 shows how easy it is to create user interface sketches. Analysts do not need to have technical user interface development skills.

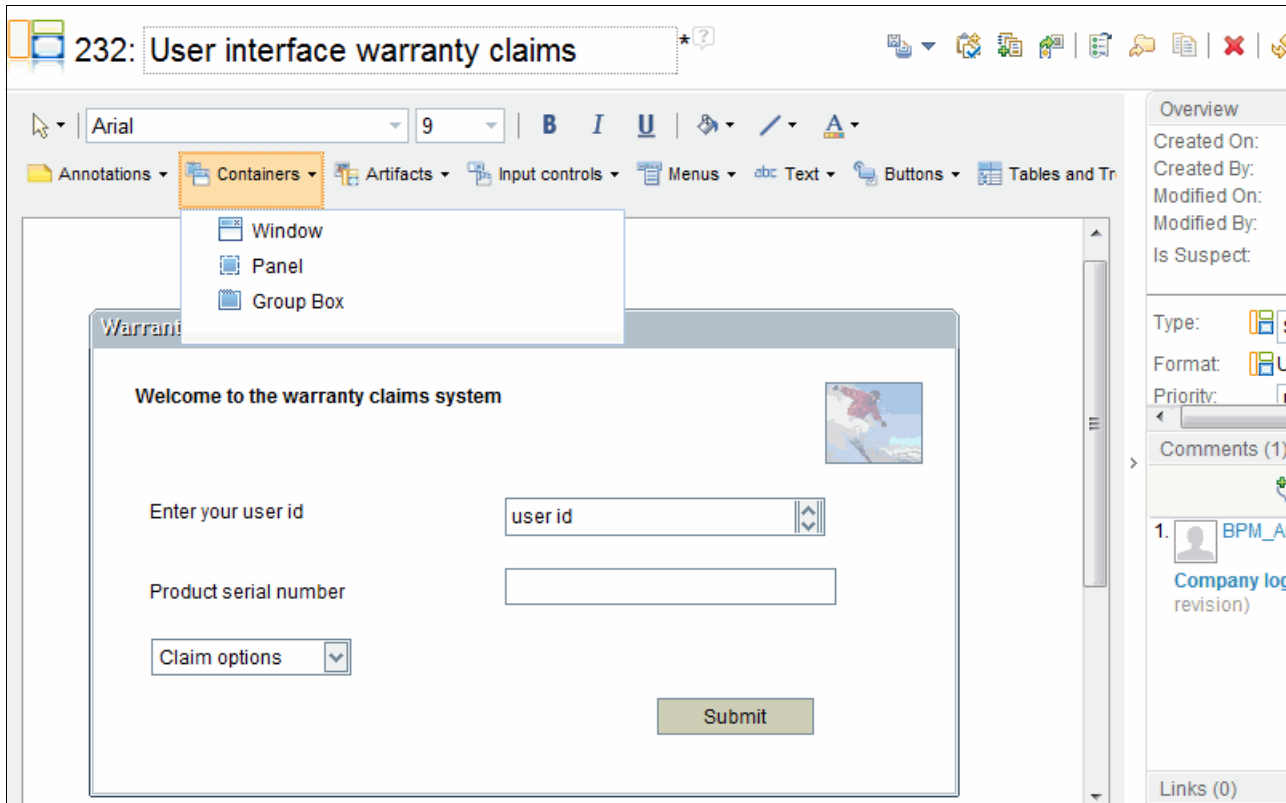


Figure 2-19 Creating a user interface sketch

Business processes can be created quickly and efficiently in the tool. Once complete, analysts can print the document as a PDF to share with stakeholders in order to confirm the flow. See Figure 2-20.

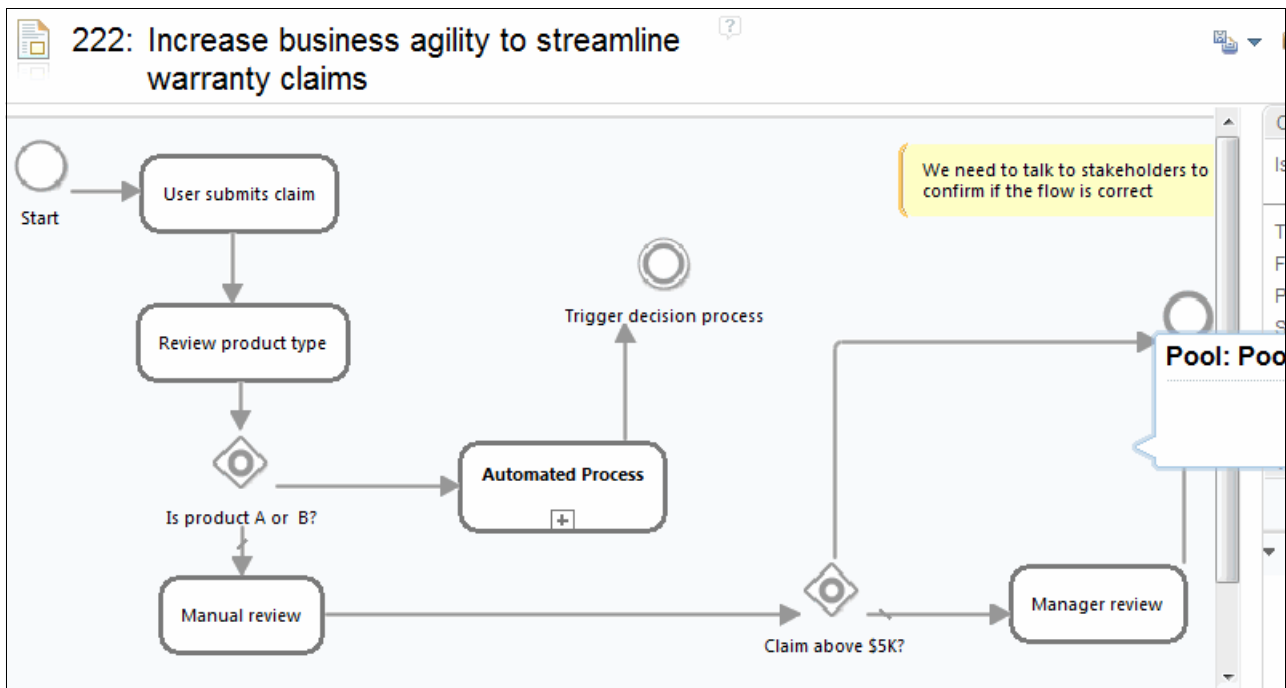


Figure 2-20 Creating a business process

Analysts can ensure that they have captured users' needs by setting up formal and informal reviews. Users can participate in the reviews, as needed, to achieve full consensus. See Figure 2-21.

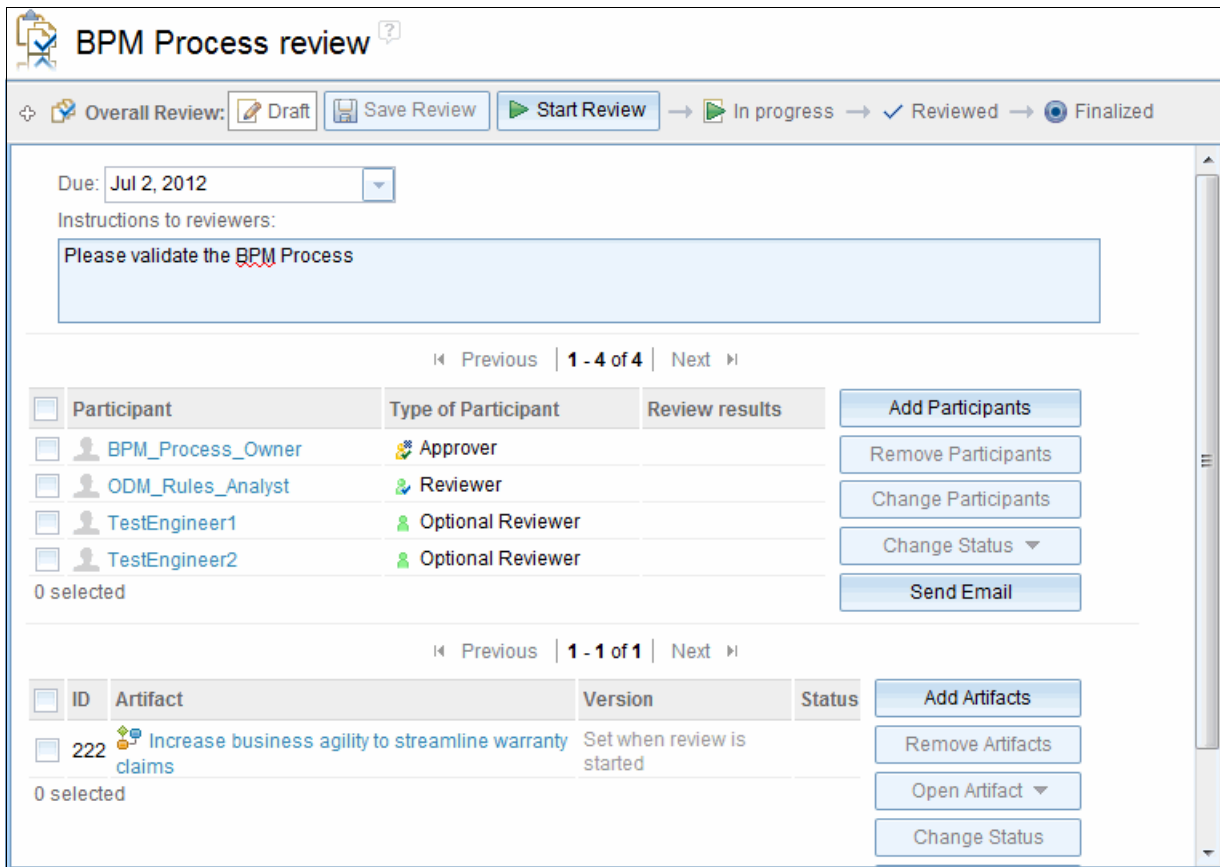


Figure 2-21 Creating a review for an artifact

When analysts get an agreement with users, they can document system features and nonfunctional requirements in rich text editors so that requirements are clearly documented. Artifacts can be linked to other requirements artifacts. See Figure 2-22 on page 22.

223: User needs to have the ability to enter warranty claims in the system

Card:
As a user of the warranty claims system I would like to enter claims in a system, so that I can register product issues and apply for the warranty.

Conversation:
User stated he needs to enter the claim in quick and easy way. We suggested a simple web page where he can enter the serial number of the product and select a reason, he agreed.

Confirmation:
Acceptance criteria:
Through a simple web page, a registered user is able to enter a warranty claim by providing only the product serial number and selecting a reason for the claim.

Overview
Type: User Story
Format: Text
Priority: Medium
Status: Approved

Links (2)
224: User interface for warranty claims

Link From (1)
224: User interface for warranty claims

Where Used

Figure 2-22 Creating a user elaboration artifact

Other users, such as the developers or test engineers, cannot change the requirements. They can, however, enter rich text comments to inform the analysts of needed changes.

When the requirements are established, analysts create a Release Collection to generate the set of requirements that are to be implemented in the release by going to **Collections** → **Create Collection** and then adding this set of artifacts.

Browsing a collection can help the team to confirm the requirements that are in plan for the release, and the artifacts can be grouped by different options to make them easier to find or to create tighter associations. Figure 2-23 on page 23 shows a group by business priority.

219: WarrantyClaimsR1

Page size: 20

No grouping

ID	Name	Artifact Type	Last Modified By	Last Modified Date	Implemented By	Validated By
220	The system should provide the ability to handle exceptions	Feature	BPM_Analyst	Jun 25, 2012 7:46:26 AM	96: As a customer I would like to enter warranty claims so that I can get a valid answer on how to proceed with the purchase product	
221	The system should provide a useful set of reports	Feature	ProjectLeader	Jun 12, 2012 11:40:03 AM		
222	Increase	User Story	BPM_Analyst	Jun 25, 2012 7:46:59		

Showing 10 of 10 Artifacts

Figure 2-23 Browsing a collection

When the development and test plans have been created, you can link these plans to the collection to keep track of how the features are developed and tested.

2.4.3 Requirements traceability

Since a business environment constantly changes, applications need to be updated quickly. When they are still under development, a change request is created and teams need to quickly understand the impact of the change given the current stage of the development process. Changes in early stages may not have as big an impact as those that occur later in the cycle.

Teams need to understand which artifacts could be impacted if the change is approved, and this impact goes beyond just updating a requirement. All code that has been developed and tested would need to be analyzed to assess the impact of the change. This reassessment would require prompt action by all teams.

Stand-alone requirement tools can help to develop a traceability matrix, but, if the development and test tools cannot feed the requirements tool continuously, it becomes hard to keep the generated matrix up to date over time. Teams can lose track of the requirements along with any changes to each. Requirements traceability helps provide the needed control over this type of change.

The IBM Solution for Collaborative Lifecycle Management helps teams keep track of all their artifacts. Dashboards provide useful viewlets to help identify which requirements are implemented by a particular development artifact or tested by a specific test case. See Figure 2-24 on page 24.

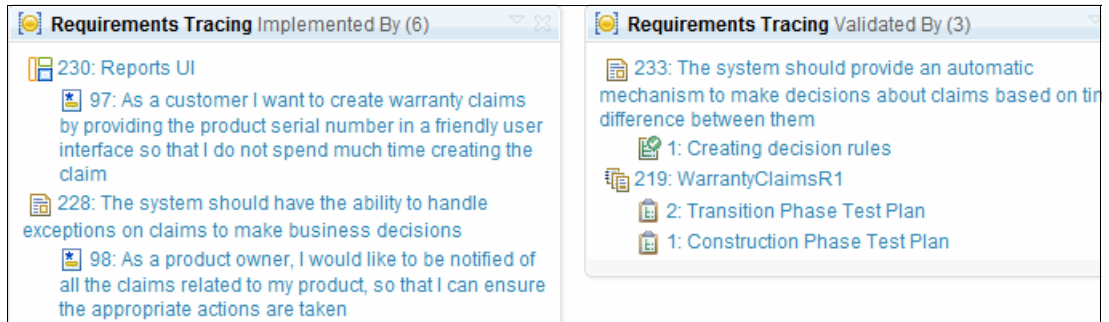


Figure 2-24 Requirements traceability

Reports can be created to share information with stakeholders who may not have access to the system. In addition, different types of reports can be created, as shown in Figure 2-25. The five steps to generate any report are:

1. Select the document-style report type.
2. Select the artifacts.
3. Report information.
4. Save the document-style report to project.
5. Generate the document-style report.

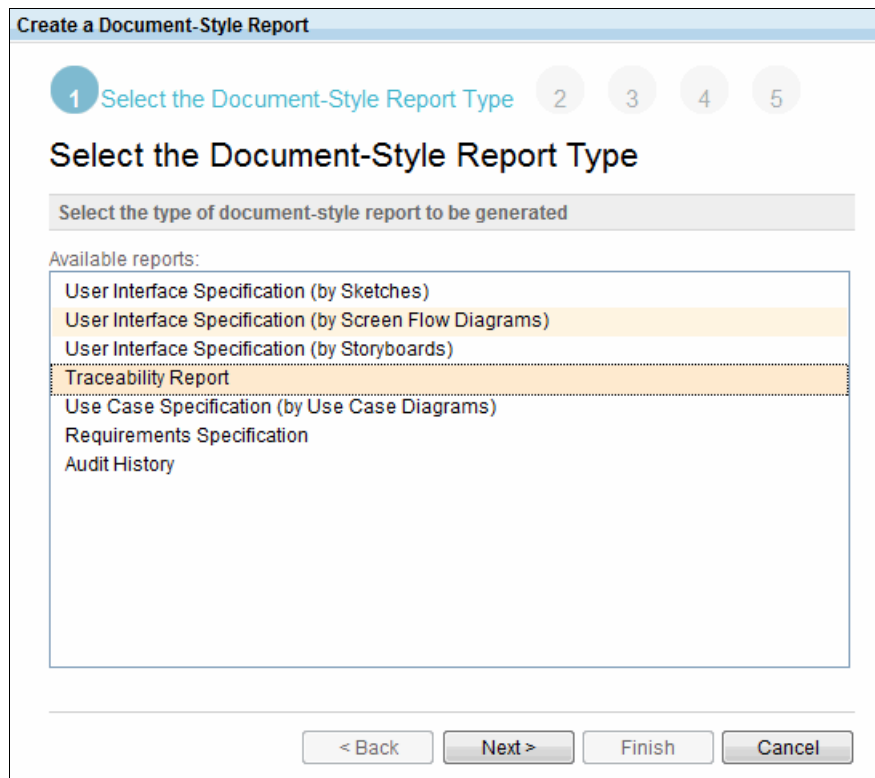


Figure 2-25 Creating requirements reports

2.5 Creating the release and iteration plans

From the inception phase and until the transition phase is completed, you and your team work in different activities. You need to plan the whole effort according to your process, and you also need a way to track the progress so that you are able to provide status to project stakeholders.

Rational Team Concert provides many functions for creating and tracking of plans and work items. A plan consists of a collection of work items that are associated with an iteration of the timeline.

2.5.1 Creating a plan

Projects are organized into different periods with fixed lengths. Within Rational Team Concert projects are organized into timelines. Each timeline consists of a set of iterations, which are defined by start and end dates. You can create different types of plans.

The following list displays the most common plan types:

- Release plan** A Release plan is a top-level plan and shows an overview of the goals of the project. Top-level work items such as user stories or use cases can be stored into this plan and can also contain a set of iteration plans.
- Iteration plan** An Iteration plan defines goals and work items that must be finished during an iteration. Iteration plans can be assigned to a project area or to a specific team separately so that each team can track their progress.

To create a plan, navigate to Plans and choose the type of plan you need. See Figure 2-26.

To complete the creation of the plan, follow these steps:

1. Give the plan a name, for instance Release plan.
2. To choose an Owner, click **Browse** and select a team area. In the case of a release plan, you can use the Project Area, since this plan will cover the work of the team as a whole. If you create iteration plans, you can attach one to each team and choose a team area for each one. For example, you can have an Elaboration Data Modeling plan attached to your Data Model team area.
3. To choose an Iteration, click **Browse** and select an iteration. In our example, since we are creating a Release plan, choose Release 1.0.
4. Click **Save**.

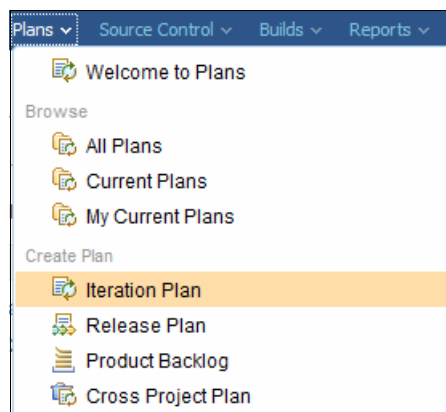


Figure 2-26 Creating a plan

The plan contains all the work items that are associated with the release or iteration you selected. You can add existing work items by associating them to the iteration, or you can create new ones. Figure 2-27 shows the release plan for the warranty claims sample.

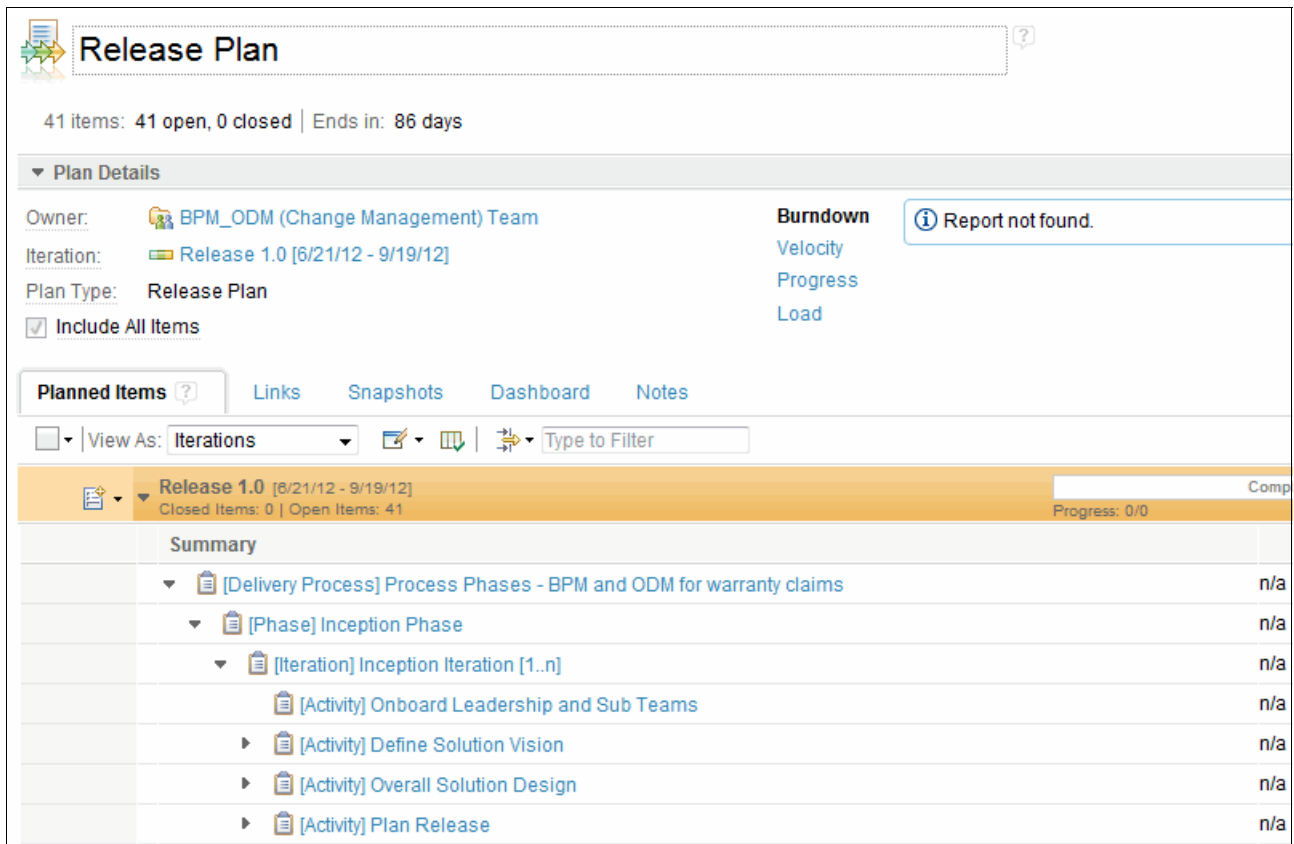


Figure 2-27 Creating a release plan

There are different views for a plan. See Figure 2-28 on page 27. The most commonly used are:

- Work Breakdown** This view shows the owners of the work items and their progress. This is a useful view for team members.
- Roadmap** The Roadmap provides a Gantt chart of all work items of a given plan. With this view you can track the accumulated time of each work item.
- Ranked list** Displays the work items based on a ranking you establish. This is very useful if you have many tasks with high priority and you need to establish which one should go first.
- Traceability** Displays the implemented requirements, tested by test case and affected by defect columns, which link a specific development work item with artifacts in the Rational Collaborative Lifecycle products.

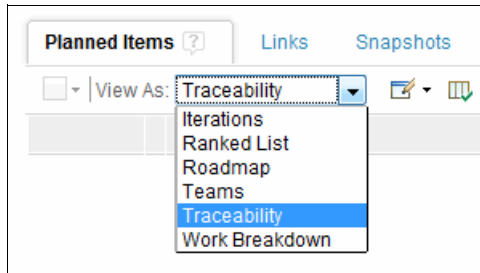


Figure 2-28 Plan views

The plan helps to track work items and these are the tasks the team needs to do to complete an effort.

The most common work items are tasks and defects. To create and add a work item to a plan follow these steps:

1. Open the menu Work Items and select the one you need.
2. Set values to the following fields:
 - a. Summary - a short description of the task.
 - b. Filed Against - project area or team area to which the task belongs.
 - c. Owned By - the user who is responsible for this task.
 - d. Planned For - the iteration in which the task is expected to be done.
3. Click **Save**.

The newly created work item is added to the plan and area automatically when you selected the values of **Planned For** and **Filed Against**.

You can link your plan to other items by going to the Links tab and selecting the appropriate option. See Figure 2-29.

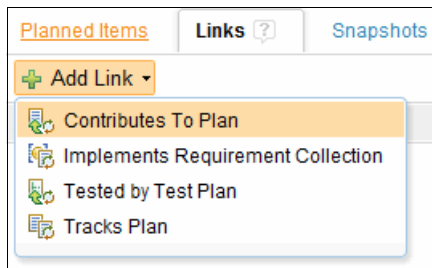


Figure 2-29 Linking the plan

To get an overview about the created plans, navigate to **Plans** → **All Plans**.

2.5.2 Creating plan snapshots

Once your plan is ready you can establish a snapshot of it in order to record the current state of the plan and the schedule and status of the associated work items. You can compare the current plan with previous snapshots to analyze the progress of a project.

To create a snapshot, follow these steps:

1. Open the plan.
2. Switch to the **Snapshots** tab.
3. Click the **Create Snapshot** link.

4. There are two types of snapshots:
 - a. Regular, which can be created at any time and is non-limited.
 - b. Planned, which can only be created once, and represents the baseline.
5. Click **OK**.

Now that you created your snapshot, you can compare it against the current plan or other snapshots. To perform the comparison, follow these steps:

1. Open the plan.
2. Switch to the Snapshot tab.
3. Select some snapshots.
4. Click **Compare Snapshots**, which is located on the right side.

The snapshot shows the changes in dates as well as the list of modified work items, added items, or removed items. See Figure 2-30.

	Comparison - Jun 25, 2012 9:50:11 AM	Current Plan - Jun 25, 2012 9:50:11 AM	Release
Overview			
Iteration Start Date	0	Jun 21, 2012	Jun 21
Iteration End Date	0	Sep 19, 2012	Sep 19
Scheduled Start Date	0	Jun 21, 2012	Jun 21
Scheduled End Date	0	Sep 19, 2012	Sep 19
Work Items Total	1	42	41
New	1	42	41
In Progress	0	0	0
Closed	0	0	0
<i>Not Accessible</i>	--	--	--
Work Done	0	0	0
Work Left	0	0	0
Progress	--	0/0 h <input type="text" value="No Work Estimated"/>	0% 0/0 h
Work Items			
	Modified Items	Added Items	Removed
	[Milestone] Stakeholder Consensus Achieved	Create the data model	--
	--	Rules not working	--

Figure 2-30 Plan snapshot

2.5.3 Working with artifacts

Work items are useful to track work and to establish relations between activities by linking artifacts. Therefore it is important to keep work items up-to-date.

We suggest that you add comments to your work items to share information regarding status and progress. Other users may subscribe to any work item in order to keep abreast of changes and to engage as necessary.

To subscribe to a particular work item you just need to click **Subscribe me** located in the top right icons section of each work item. You can also go to the **Link** tab of each work item and add yourself or other users. See Figure 2-31 on page 29.



Figure 2-31 Subscribing users to work items

To update the status of a work item, open it and change the status using the drop-down menu on the right side, as shown in Figure 2-32. Then save your changes by clicking **Save**.



Figure 2-32 Change the status of a work item

To add a link to a work item, follow these steps:

1. Open a work item.
2. Go to the **Links** tab.
3. Click **Add**.
4. Select one link type. For example:
 - a. **Implements Requirement** - To associate the work item with the requirement for which it is currently being implemented.
 - b. **Related Test Case** - To link a test case to the work item.
 - c. **Depends on** - To show dependencies to another work item.
 - d. **Children** - To add another work item as child of the current work item.
5. Click **Save**.

When the plan and all the work items have been created, assigned and planned, you are ready to start working on your application.

Plans and work items can be tracked by using the Rational Team Concert Eclipse client as well. Developers can also use it as an integrated development environment. Developers can commit their code and track their work items and defects easily.

Rational Team Concert provides other useful functionality not explored in this chapter, such as source code and continuous builds. For more information about Rational Team Concert go to:

<https://jazz.net/products/rational-team-concert>

If your application involves different technologies, you can also review the functionality provided by IBM Rational Application Developer, which is an Eclipse-based development environment that helps you design, develop, analyze, profile and deploy high quality Java, Java EE, portal, Web 2.0, Mobile, OSGi, and SOA applications. Rational Application Developer integrates with Rational Team Concert and is optimized for IBM WebSphere Application Server.

For more information about Rational Application Developer, go to:

<http://www-01.ibm.com/software/awdtools/developer/application/>



Data modeling with InfoSphere Data Architect

Data modeling is a method of defining the data structure, hierarchy, relationships, and maintenance that can directly influence every aspect of information. Since it can impact both the application design and performance, data modeling is a primary structural element in information technology solutions.

Business applications are usually about processing data and are almost entirely dependent on a data model. Regardless of the architecture, a business application sits on top of a data model or other type of data structure technology. Business applications can involve many key areas, such as procurement, purchasing, invoicing, marketing, and finance. In addition, many applications must cover related day-to-day user activities, such as creating an order, making a payment or a credit, creating forecasts, or reporting on the company's business health and overall position.

In this chapter we explain how to create a data model that supports the data requirements for the sample company's warranty solution.

The following topics are included:

- ▶ Data modeling and IBM InfoSphere Data Architect
- ▶ Creating the IBM InfoSphere Data Architect workspace
- ▶ Creating the data design project
- ▶ Creating the logical data model
- ▶ Creating the physical data model
- ▶ Importing data into the WARRANTY database
- ▶ Exporting the WARRANTY data model to Cognos

3.1 Data modeling and IBM InfoSphere Data Architect

Data modeling is the stage of an information system design that is accomplished by applying techniques and tools to create a common data structure. It is usually the first step of a project involving either a database requirement or a requirement to store data. The main objective of data modeling is that a model contains *entities* and *relationships* in order to represent the specifications of a business information scenario.

Basically the data model bridges the gap between IT and business users. It organizes the way of thinking about data and can help demonstrate the significance and practical application of data. A data model establishes the link between user requirements and the IT solution. It supports the design of the application's structure and manipulates the data. In addition, from a user's perspective, the data model facilitates how to abstract and understand the system applications in an easy manner.

IBM InfoSphere Data Architect is a powerful and comprehensive enterprise data modeling tool that defines, relates, and integrates data assets in developing database applications. It provides the following features:

- ▶ Logical, physical, storage, dimensional, domain, and integration data modeling
- ▶ Visual modeling through information engineering (IE) data diagrams and topology diagrams
- ▶ Preferred practice validation for models and deployed databases, including naming standards, syntax, compliance, and normalization
- ▶ Lifecycle management functionality, including impact analysis for models, extended comparison, and synchronization functionality
- ▶ Multi-user support, including support for partitioning, cross-model references, and source code management system integration
- ▶ A unique mapping editor that helps to relate different data structures to each other with discovery capabilities to the ability to detect relationships
- ▶ Database code designers for SQL statements, stored procedures, and user-defined functions
- ▶ Support for running statements against the database

The examples in this book use IBM InfoSphere Data Architect Version 8.1. To learn more about this software, go to:

<http://pic.dhe.ibm.com/infocenter/dataarch/v8r1/nav/1>

3.2 Creating the IBM InfoSphere Data Architect workspace

Before you continue: IBM InfoSphere Data Architect must be installed and running on your workstation to complete this step. We do not discuss how to install the software here, but we encourage you to click the following link to learn more about this process:

http://pic.dhe.ibm.com/infocenter/dataarch/v8r1/topic/com.ibm.datatools.rda.install.doc/topics/c_install_product.html

To create a new IBM InfoSphere Data Architect workspace:

1. Select **Start** → **All Programs** → **IBM InfoSphere** → **IBM InfoSphere Data Architect 8.1.0.0** to open the Workspace Launcher (Figure 3-1).

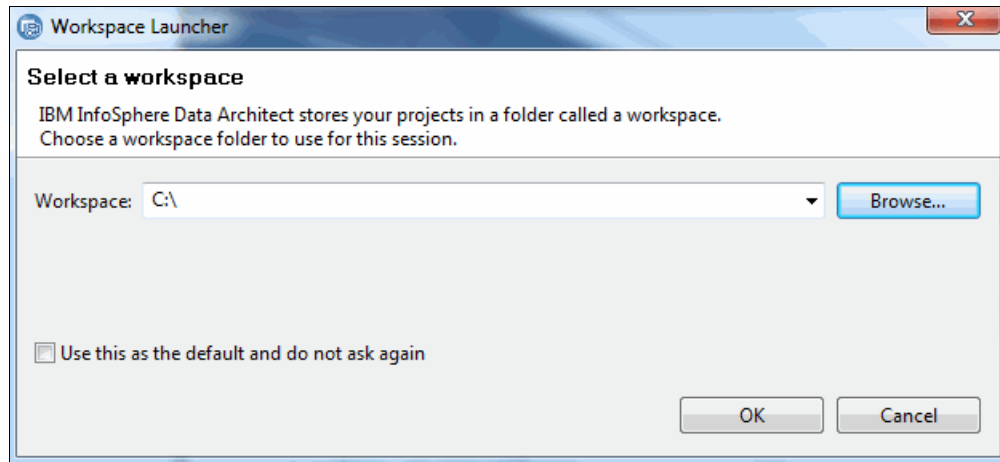


Figure 3-1 Workspace Launcher

2. Select a directory to create a new folder to store your IBM InfoSphere Data Architect workspace (Figure 3-2), and click **OK** when done.

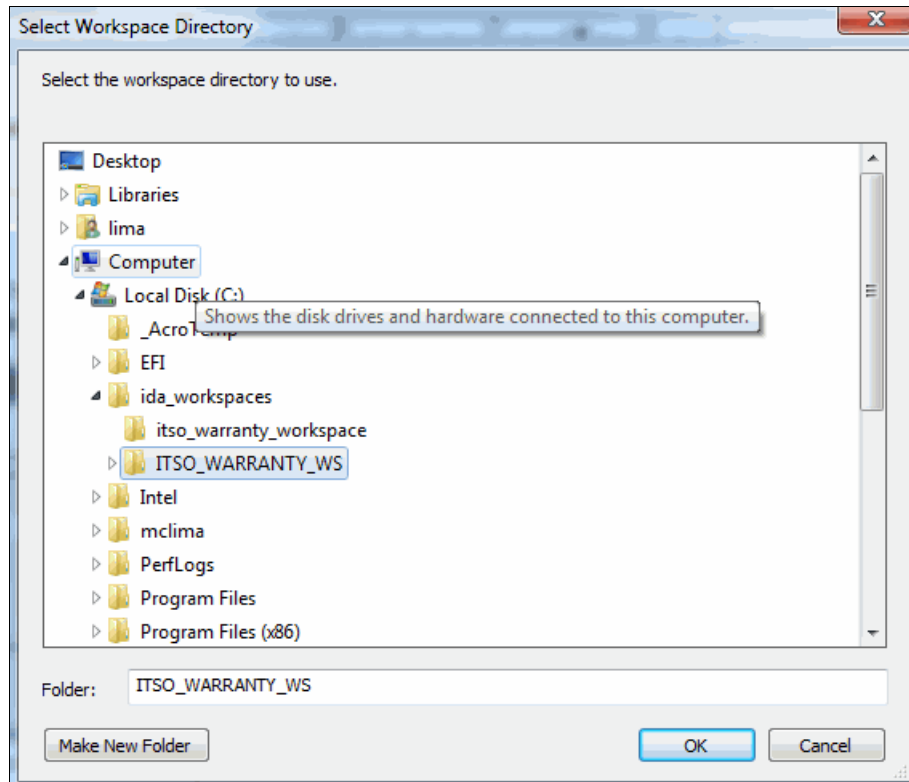


Figure 3-2 Creating a new IBM InfoSphere Data Architect workspace

3.3 Creating the data design project

The first step in creating a data modeling project is to create a *data design project*, where you will store your project objects, such as hierarchical modeling, in a meaningful way.

To create a data design project:

1. Select **File** → **New** → **Data Design Project** from the workbench toolbar (Figure 3-3).

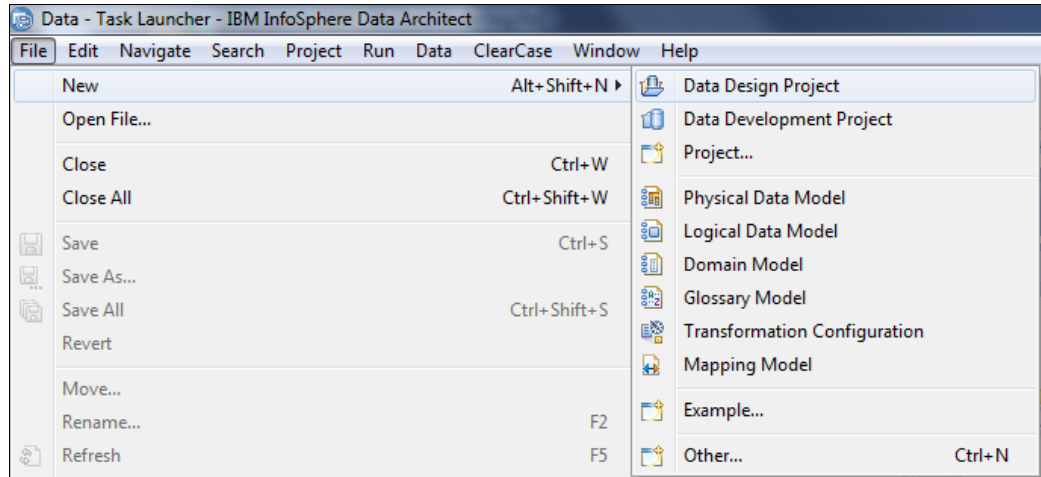


Figure 3-3 Create a new data design project

2. Specify the project name (Figure 3-4), and then click **Finish**.

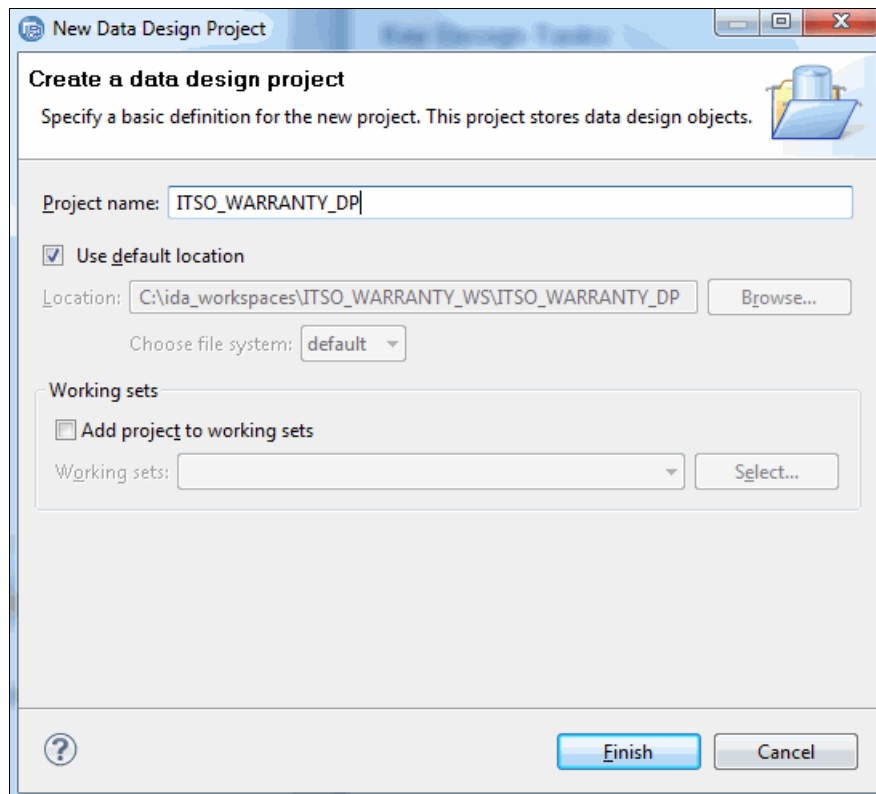


Figure 3-4 Specify the project name

3. Your new data design project will be created in the Data Project Explorer (Figure 3-5); expand ITSO_WARRANTY_DP to view the object types for this project.

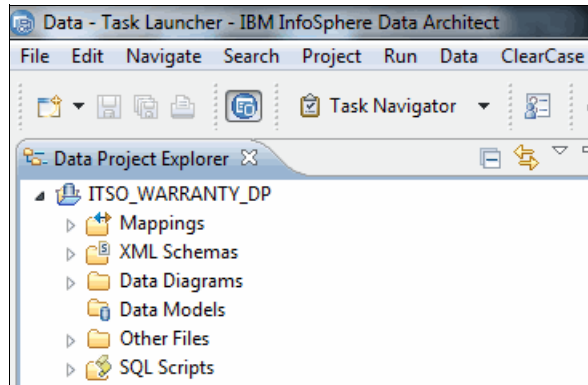


Figure 3-5 The ITSO_WARRANTY_DP entry in the Data Project Explorer

Tip: You can also create a data design project using the New Data Design Project wizard. Right-click in any blank area in the Data Project Explorer and select **New** → **Data Design Project**.

IBM InfoSphere Data Architect stores the following object types in a data design project:

- ▶ Logical data models
- ▶ Physical data models
- ▶ Domain models
- ▶ Glossary models
- ▶ SQL scripts, including Data Definition Language (DDL) scripts and stored procedures
- ▶ Web services
- ▶ Unified Modeling Language (UML) projects and diagrams
- ▶ Mapping models and XML schemas
- ▶ Transformation configuration files

The left side of the workbench window includes the Data Project Explorer and the Data Source Explorer perspectives. You can use the Data Project Explorer to view the entire data design project, while the Data Source Explorer allows you to validate the configuration repositories and database connections.

3.4 Creating the logical data model

In this section we describe how to create the logical data model for our scenario.

3.4.1 What a logical data model is

When working on an application design, you will want to understand the business requirements for the design and to translate those requirements into a concrete and understandable representation of the scenario. A *logical data model* is the first phase in this data design lifecycle. The logical data model provides a good method to refine business requirements, show those requirements in a user-friendly picture, and view how future database designs can be implemented. It is important to provide a logical data model before designing the physical storage in a database, such as DB2 or Informix®.

Before creating a logical data model, you need to clarify as many of the model requirements as possible. A business user's description might be incomplete or lacking information. Often, business users provide only a general understanding of the business process in which they are interested. Architects can provide a deeper understanding of these business processes in order to collect missing information in order to construct a complete model and deliver the value that is requested.

A logical data model is not linked to a database tool or technology. It describes what an organization or a company wants to gather data about and documents the relationships between this data. Logical data models are organized hierarchically and contain *objects* such as packages, entities, attributes, and other relationships.

3.4.2 Creating the WARRANTY logical data model

Logical data model objects are always set under a root package object, of which there is only one. You can add additional packages under the root package to group similar objects. In this example the root package is the data design project created in 3.3, "Creating the data design project" on page 34.

Now you will create the WARRANTY logical data model.

Create a logical data model

First you need to create a logical data model:

1. From the toolbar, select **File** → **New** → **Logical Data Model** (Figure 3-6)

Note: You can access these options by right-clicking in any empty area of your workbench.

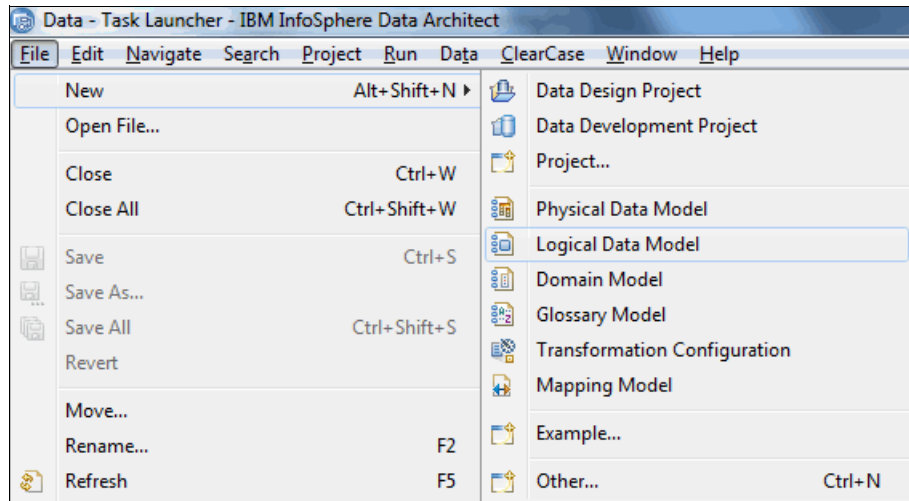


Figure 3-6 New logical data model

The New Logical Data Model wizard opens. Create your model using the blank logical data model template. If you created other templates previously, they would have been available for you to select as the basis for this logical model. You can enable the template for selection by specifying the location of the template file in the Template folder field.

2. Next, you need to name your data model. We suggest naming it `ITSO_WARRANTY_Logical_Data_Model`, as shown in Figure 3-7. When done, click **Finish**.

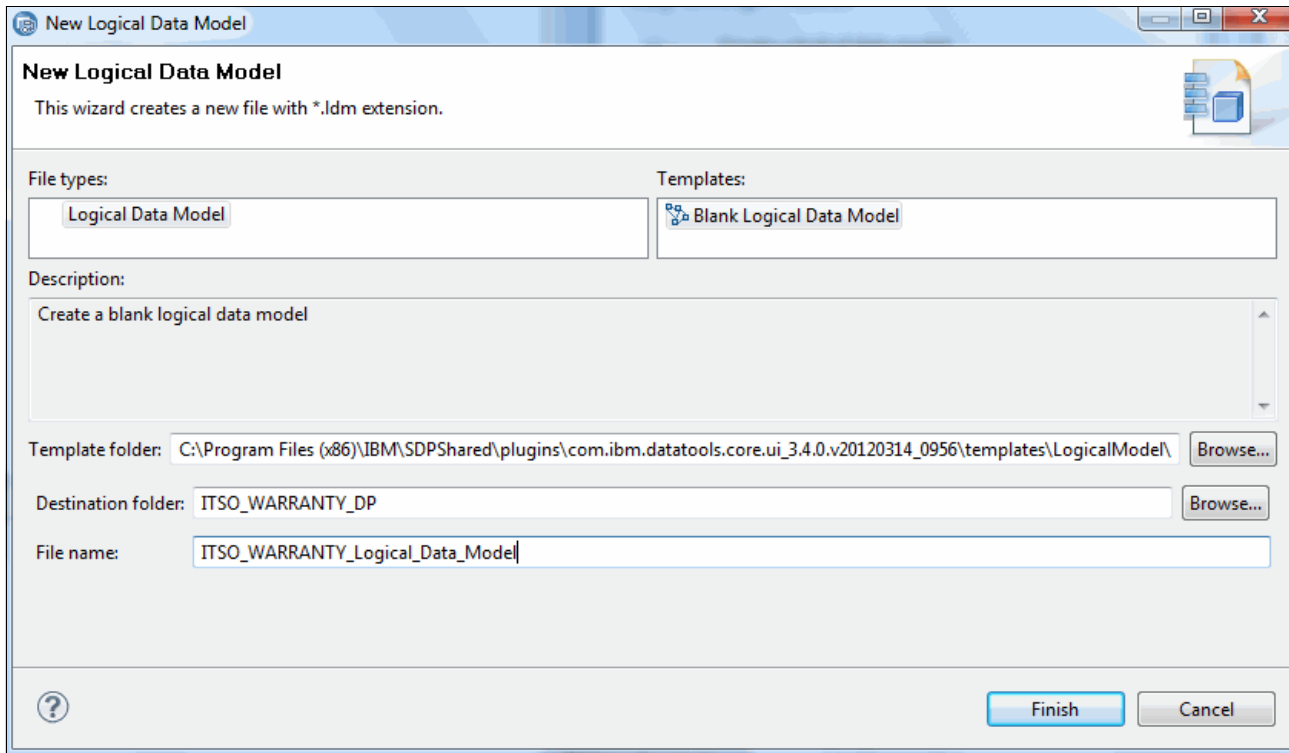


Figure 3-7 New logical data model

3. The logical data model is created under the data design project. Should you want to name your logical data object properties, right-click the logical data model object in the Data Project Explorer and select **Rename**. After entering a suitable name in the New Resource Name field, click **OK** to save your logical data model.

Create entities and attributes in the diagram

Now create entities in the diagram and define attributes for each:

- ▶ An *entity* is a logical data model object that stores information about areas that are of interest to business users.
- ▶ Each entity contains *attributes* that describe details about the entity.

For example, you can create a CUSTOMER INFO entity that contains the CUSTOMER ID, CUSTOMER FIRST NAME, and CUSTOMER LAST NAME attributes, which provide details about a customer. When the logical data model is transformed into a physical data model, these entities become *tables* and the attributes become *columns*.

Add entities to the diagram

To add an entity to your logical data model diagram:

1. Expand the Data Models folder to expose your logical data model diagram (Figure 3-8).

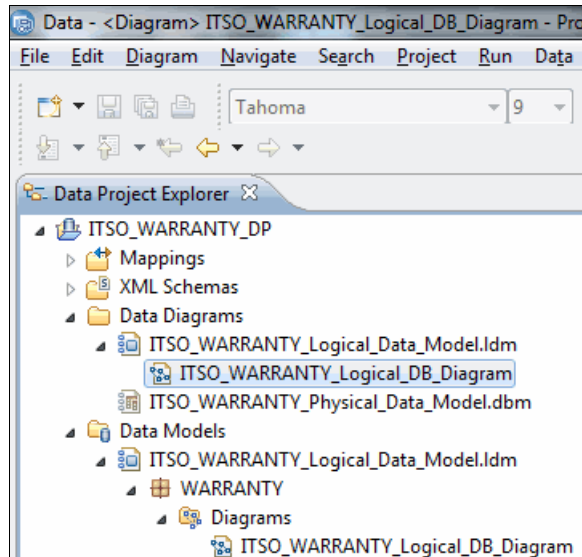


Figure 3-8 Open the logical diagram

2. Click **Entity** in the palette, and then provide the location in the diagram where you want to add it.
3. You will want to name this entity something meaningful, such as PRODUCT INFO or CUSTOMER INFO.

Add attributes to an entity

An entity can have attributes, such as primary keys and non-key attributes. To add attributes to an entity:

1. Select the entity in the diagram to open the Properties view.
2. Select the Attributes tab under the Properties view, and then click **New** to add your attribute.
3. You can also adjust other properties, such as Name, Primary Key, or Type, from the Properties view.

Create the entities for the WARRANTY logical data model

Now, create the entities for the WARRANTY logical data model. In the palette view, expand the Data section. Next, drag six entities into any blank area in the diagram editor and name each as follows:

- ▶ CUSTOMER INFO
- ▶ PRODUCT INFO
- ▶ WARRANTY LEVEL INFO
- ▶ WARRANTY STATUS INFO
- ▶ CUSTOMER PRODUCT SALES
- ▶ WARRANTY TRANSACTIONS

Figure 4-9 shows how the entities look after you created them.



Figure 3-9 The WARRANTY logical data model entities

Add attributes for each entity

Next, add attributes for each entity:

1. Select the CUSTOMER INFO entity to open the Properties view, as shown in Figure 3-10.

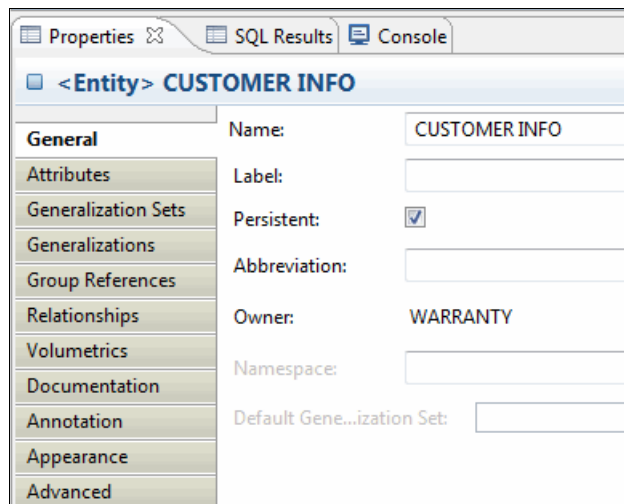


Figure 3-10 CUSTOMER INFO entity properties

2. Click the Attributes tab and then click **New** (Figure 3-11).

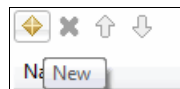


Figure 3-11 New icon

A new attribute is created in the Properties view and the diagram, as shown in Figure 3-12.

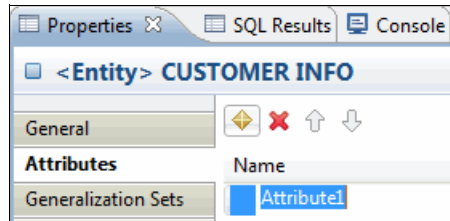


Figure 3-12 New CUSTOMER INFO attribute

3. Change the properties as follows:

- Name: CUSTOMER ID
- Type: INTEGER

Select the **Primary key** option.

Repeat these steps for the remaining attributes shown in Figure 3-13.

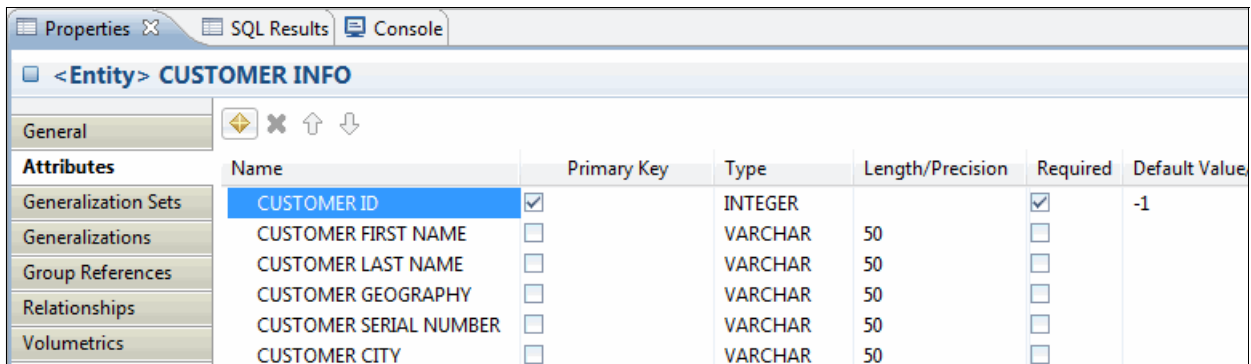


Figure 3-13 The CUSTOMER INFO attributes

To create the PRODUCT INFO attributes, repeat these steps for the attributes shown in Figure 3-14.

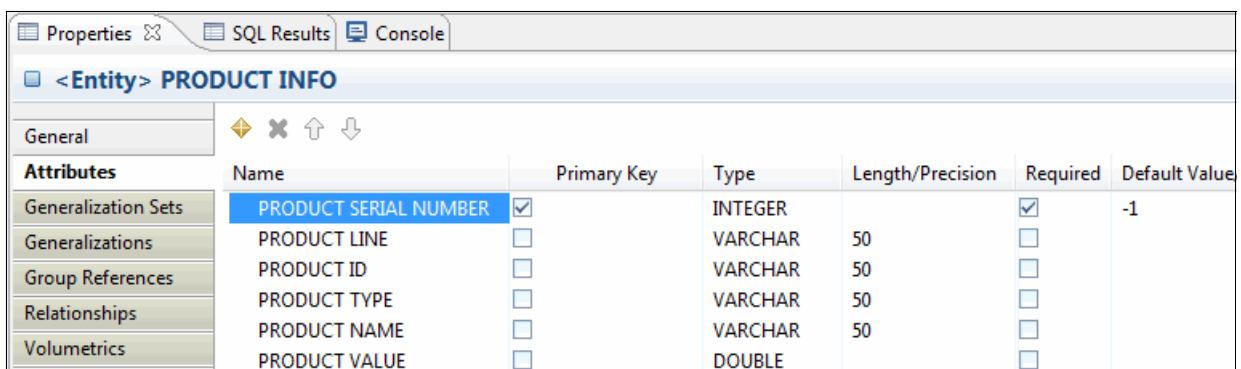
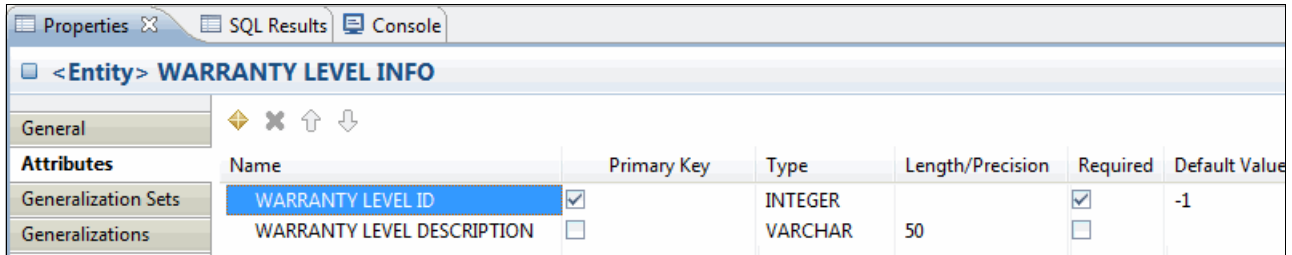


Figure 3-14 The PRODUCT INFO attributes

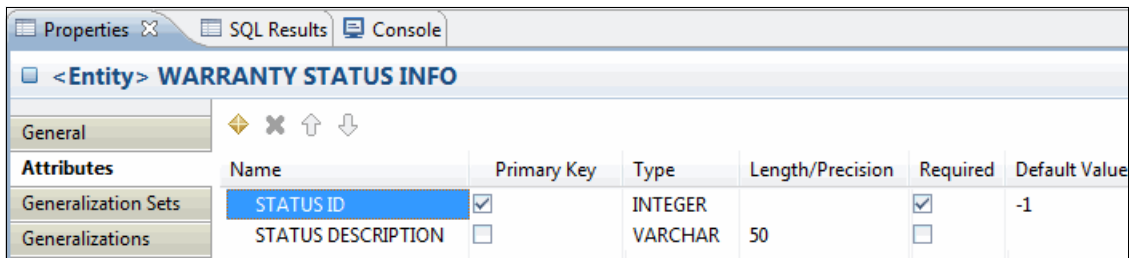
To create the WARRANTY LEVEL INFO attributes, repeat these steps for the attributes shown in Figure 3-15.



<Entity> WARRANTY LEVEL INFO						
General						
Attributes						
	Name	Primary Key	Type	Length/Precision	Required	Default Value
Generalization Sets	WARRANTY LEVEL ID	<input checked="" type="checkbox"/>	INTEGER		<input checked="" type="checkbox"/>	-1
Generalizations	WARRANTY LEVEL DESCRIPTION	<input type="checkbox"/>	VARCHAR	50	<input type="checkbox"/>	

Figure 3-15 The WARRANTY LEVEL INFO attributes

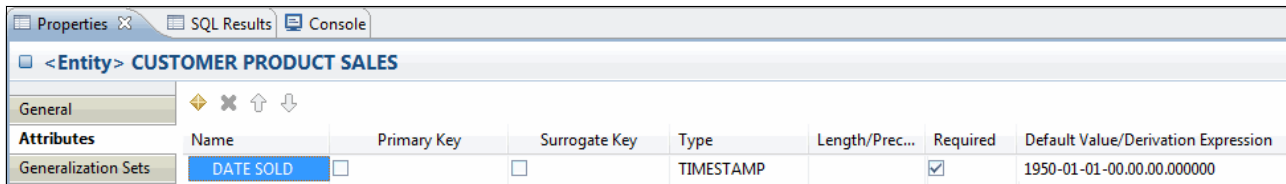
To create the WARRANTY STATUS INFO attributes, repeat these steps for the attributes shown in Figure 3-16.



<Entity> WARRANTY STATUS INFO						
General						
Attributes						
	Name	Primary Key	Type	Length/Precision	Required	Default Value
Generalization Sets	STATUS ID	<input checked="" type="checkbox"/>	INTEGER		<input checked="" type="checkbox"/>	-1
Generalizations	STATUS DESCRIPTION	<input type="checkbox"/>	VARCHAR	50	<input type="checkbox"/>	

Figure 3-16 The WARRANTY STATUS INFO attributes

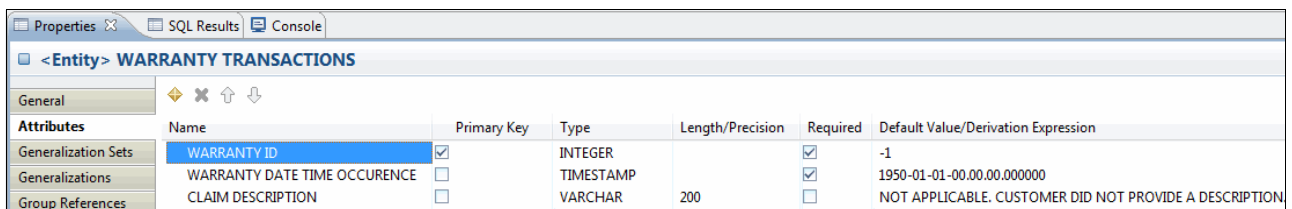
To create the CUSTOMER PRODUCT SALES attributes, repeat these steps for the attributes shown in Figure 3-17.



<Entity> CUSTOMER PRODUCT SALES							
General							
Attributes							
	Name	Primary Key	Surrogate Key	Type	Length/Prec...	Required	Default Value/Derivation Expression
Generalization Sets	DATE SOLD	<input type="checkbox"/>	<input type="checkbox"/>	TIMESTAMP		<input checked="" type="checkbox"/>	1950-01-01-00.00.00.000000

Figure 3-17 The CUSTOMER PRODUCT SALES attributes

To create the WARRANTY TRANSACTIONS attributes, repeat these for the attributes shown in Figure 3-18.



<Entity> WARRANTY TRANSACTIONS						
General						
Attributes						
	Name	Primary Key	Type	Length/Precision	Required	Default Value/Derivation Expression
Generalization Sets	WARRANTY ID	<input checked="" type="checkbox"/>	INTEGER		<input checked="" type="checkbox"/>	-1
Generalizations	WARRANTY DATE TIME OCCURENCE	<input type="checkbox"/>	TIMESTAMP		<input checked="" type="checkbox"/>	1950-01-01-00.00.00.000000
Group References	CLAIM DESCRIPTION	<input type="checkbox"/>	VARCHAR	200	<input type="checkbox"/>	NOT APPLICABLE. CUSTOMER DID NOT PROVIDE A DESCRIPTION

Figure 3-18 The WARRANTY TRANSACTIONS attributes

Tip: When you set an attribute as a primary key, the Required property is enabled automatically, because a primary key can never be a null value.

Create relationships

Now that you have created the WARRANTY entities, you can specify how the entities are related. You can specify the following types of relationships between data objects using the diagram editor:

- ▶ Identifying

In an *identifying* relationship, the existence of the child entity is dependent on the parent relationship. For example, an EMPLOYEE data object can have an identifying relationship with a WORKCOMPLETED data object.

- ▶ Non-Identifying: Optional, Mandatory, or One-to-One

Non-identifying relationships are relationships between two independent entities. For example, a CUSTOMER data object can have a non-identifying relationship with an INVOICE data object. You can specify whether these relationships are enforced (mandatory) or not enforced (optional). You can also specify a one-to-one relationship, where each row in one entity is related to only one row in the other entity.

- ▶ Many-to-many

This type of relationship is between two entities in which each row in one entity can be related to more than one row in the other entity. For example, each person in an EMPLOYEE table can have multiple projects in a PROJECT table, and multiple people can work on a project.

To create a foreign key relationship:

1. In the palette, under the Data Section, select one of the relationships.
2. Select the parent entity that has the primary key and drag it to the child entity.

Note: Depending on the type of relationship that you are creating, a window might open prompting you to specify relationship options.

3. Under the Details tab of the Properties panel, change the Verb phrase and the Inverse verb phrase to values that better express the relationship between the parent and child entity that you are creating.

After you create the relationship, you can design attributes and relationships. In this example, you design attributes and relationship for the CUSTOMER PRODUCT SALES foreign key relationship. Follow these steps:

1. In the palette, under the Data Section, select the Identifying relationship.
2. Select the CUSTOMER INFO parent entity that has the primary key and drag that parent entity into the CUSTOMER PRODUCT SALES child entity; the foreign key will be created within the CUSTOMER PRODUCT SALES child entity.
3. Under the Details tab of the Properties panel for this relationship, change the Verb phrase to buys and the Inverse verb phrase to is sold for.
4. In the palette, under Data Section, select the Identifying relationship.
5. Select the PRODUCT INFO parent entity that has the primary key and drag it to the CUSTOMER PRODUCT SALES child entity; the foreign key is created within the CUSTOMER PRODUCT SALES child entity.
6. Under the Details tab of the Properties panel for this relationship, change the Verb phrase to belongs to and the Inverse verb phrase to is composed by.

Figure 3-19 on page 43 shows the relationships between the CUSTOMER PRODUCT SALES child entity and CUSTOMER INFO and PRODUCT INFO parent entities.

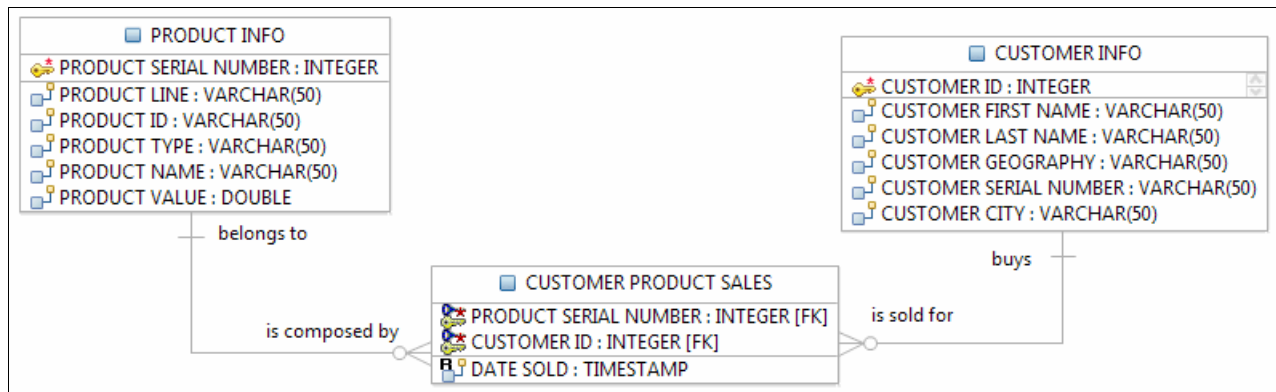


Figure 3-19 The CUSTOMER PRODUCT SALES relationships

Tip1: Depending on the type of relationship that you are creating, you might need to specify relationship options.

Tip2: Set the default value of the CUSTOMER ID and PRODUCT SERIAL NUMBER attributes to -1.

You can also design foreign key attributes and relationships for the SALES WARRANTY TRANSATIONS child entity:

1. In the palette, under the Data Section, select the Non-Identifying Mandatory relationship.
2. Select the CUSTOMER PRODUCT SALES parent entity that has the primary key and drag the parent entity into the SALES WARRANTY TRANSATIONS child entity; the foreign key is created within the SALES WARRANTY TRANSATIONS child entity.
3. Under the Details tab of the Properties panel for this relationship, change the Verb phrase to claims and the Inverse verb phrase to is claimed for.
4. In the palette, under the Data Section, select the Non-Identifying Mandatory relationship.
5. Select the WARRANTY LEVEL INFO parent entity that has the primary key and drag the parent entity into the SALES WARRANTY TRANSATIONS child entity; the foreign key is created within the SALES WARRANTY TRANSATIONS child entity.
6. Under the Details tab of the Properties panel for this relationship, change the Verb phrase to determines and the Inverse verb phrase to is determined by.
7. In the palette, under the Data Section select the Non-Identifying Mandatory relationship.
8. Select the WARRANTY STATUS INFO parent entity that has the primary key and drag the parent entity into the SALES WARRANTY TRANSATIONS child entity; the foreign key is created within the SALES WARRANTY TRANSATIONS child entity.
9. Under the Details tab of the Properties panel for this relationship, change the Verb phrase to defines and the Inverse verb phrase to is defined by.

Figure 3-20 on page 44 shows the relationship between the WARRANTY TRANSACTIONS child entity and CUSTOMER PRODUCT SALES, WARRANTY LEVEL INFO, and WARRANTY STATUS INFO parent entities.

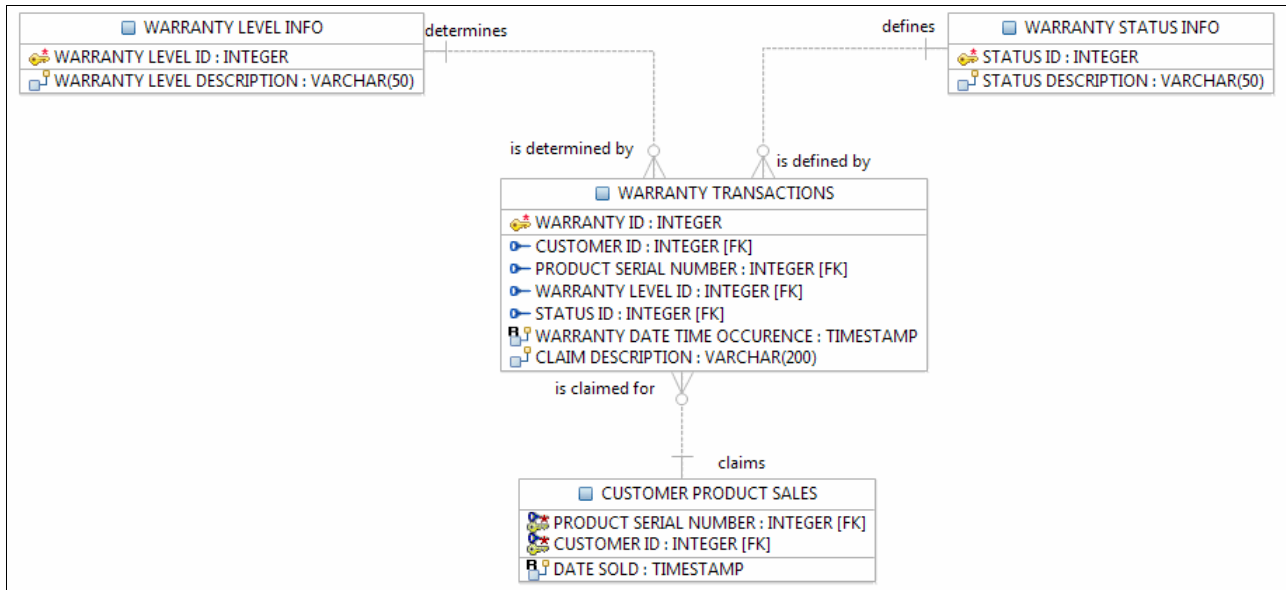


Figure 3-20 WARRANTY TRANSACTIONS relationships

Tip: Depending on the type of relationship that you are creating, you might be prompted to specify relationship options. Set the default value of the CUSTOMER ID, PRODUCT SERIAL NUMBER, WARRANTY LEVEL ID, and STATUS ID attributes to -1.

Figure 3-21 shows the complete WARRANTY logical data model, which provides a database agnostic view of what is required by the warranty solution and the business users.

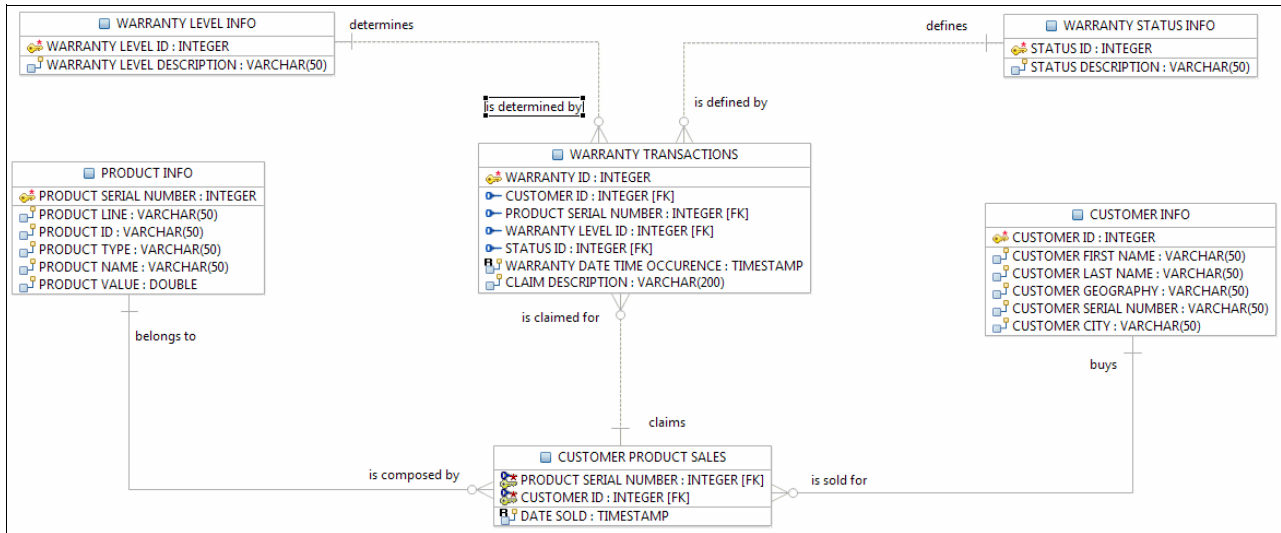


Figure 3-21 WARRANTY logical data model

3.5 Creating the physical data model

In this section we describe how to create the physical data model for our scenario.

3.5.1 What is a physical data model

A *physical data model* is a database-specific model that represents relational data objects, (for example tables, columns, primary keys, foreign keys, and data types), and how they are connected and dependent on each other.

IBM InfoSphere Data Architect can create a physical data model in several ways:

- ▶ From a wizard, creating a blank physical data model
- ▶ Using a wizard and creating a blank physical data model from a template
- ▶ Using the Data Model Import Wizard and importing a physical data model
- ▶ Reverse engineer a physical data model from a database
- ▶ Using a wizard and creating a database from a DDL file

Note: Data Definition Language (DDL) is a subset of SQL. It is a language for describing data and its relationships in a database.

When you generate DDL, you can use the DDL statements to recreate everything about a database except for its contents. You can generate the DDL to completely recreate the database, or choose to recreate only certain aspects of it, such as its current statistics. You can also limit the statements that are generated so that only a segment of the database is recreated, for example the statistics for a subset of tables.

3.5.2 Transforming a logical data model to a physical data model

Before you begin: In this section you will use the logical data model created in 3.4.2, “Creating the WARRANTY logical data model” on page 36 to build your physical data model.

Transforming an existing data model allows you to create the physical design of a database or schema automatically, which can save some of the manual work, such as creating tables, columns, and other data objects.

You can use the Transformation Wizard in IBM InfoSphere Data Architect to automate the initial physical design creation. However, keep in mind that many databases have length limits for table and column names. These limits can affect the conversion process when running DDL to create the database, because entity and attribute names might exceed the limits that are imposed by database management systems. In these cases, you will need to validate that all standard abbreviations or shortened names are assigned properly. Such procedures can be executed without a modeling tool, but the amount of manual effort will be greater.

Using IBM InfoSphere Data Architect allows you to make changes more easily and keep the logical and physical models synchronized whenever your business processes and data requirements evolve or change.

To create the WARRANTY physical data model:

1. Right-click **ITSO_WARRANTY_Logical_Data_Model** and select **Transform to physical data model** (Figure 3-22 on page 46).

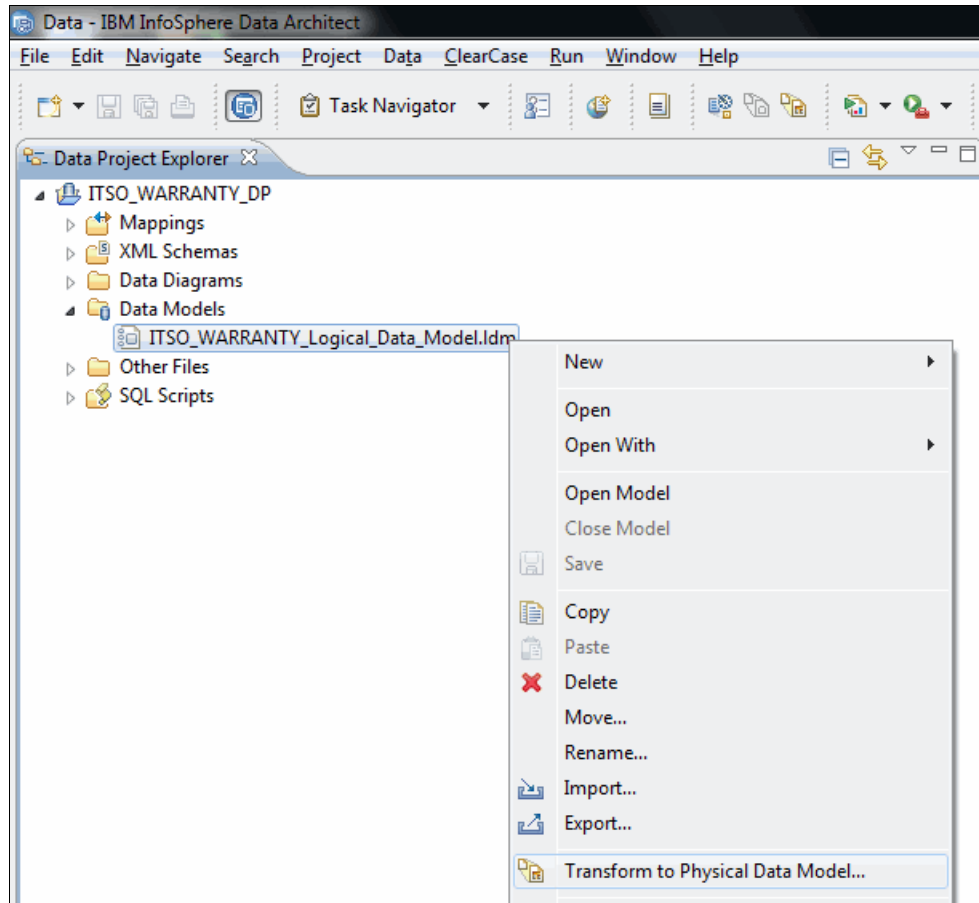


Figure 3-22 Transforming logical data model to a physical data model

2. Select **Create new model** (Figure 3-23) and click **Next**.

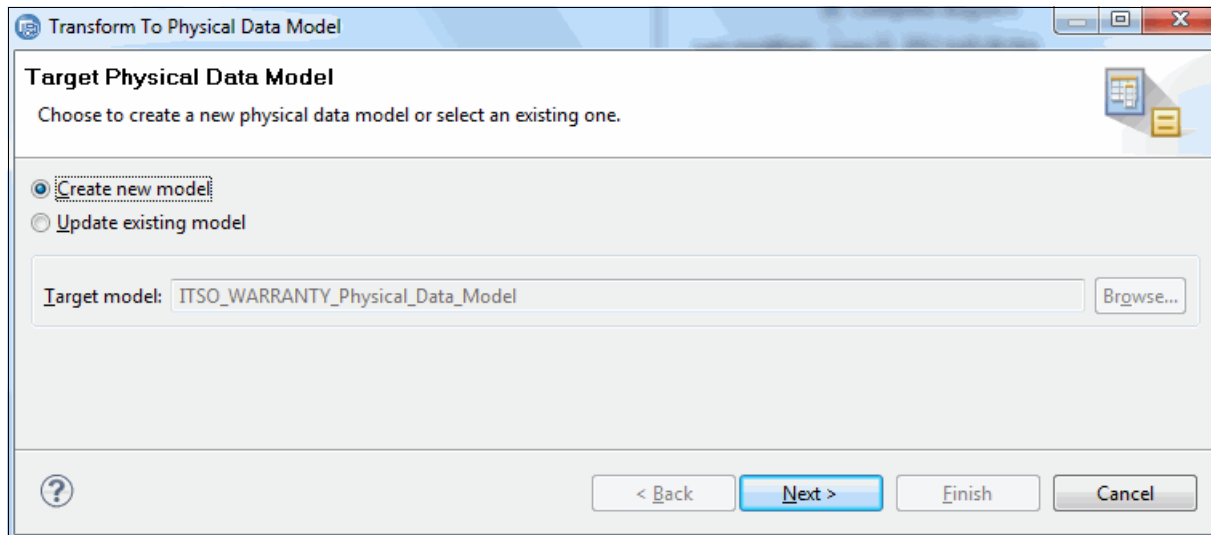


Figure 3-23 Create new physical data model

3. Enter a name, such as `ITSO_WARRANTY_Physical_Data_Model` in the “File name” field, select the database type and version, and click **Next** when finished (see Figure 3-24.)

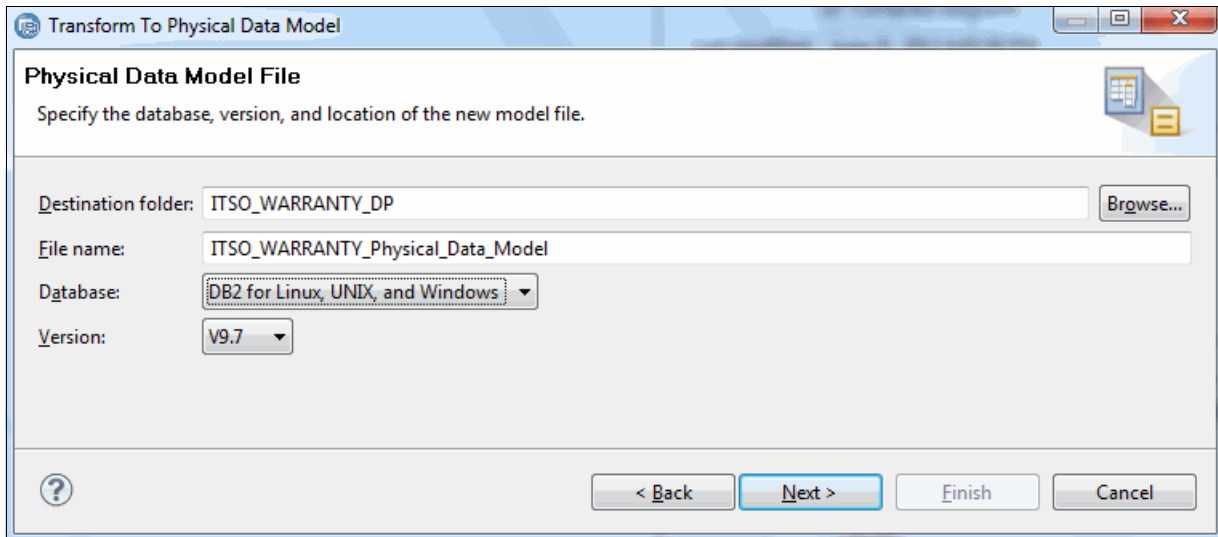


Figure 3-24 Provide information about the physical data model

4. Leave the default options unchanged in the Transform To Physical Data Model dialog that opens, but enter a name, such as WARRANTY, in the “Schema name” field and click **Next** when done (see Figure 3-25).

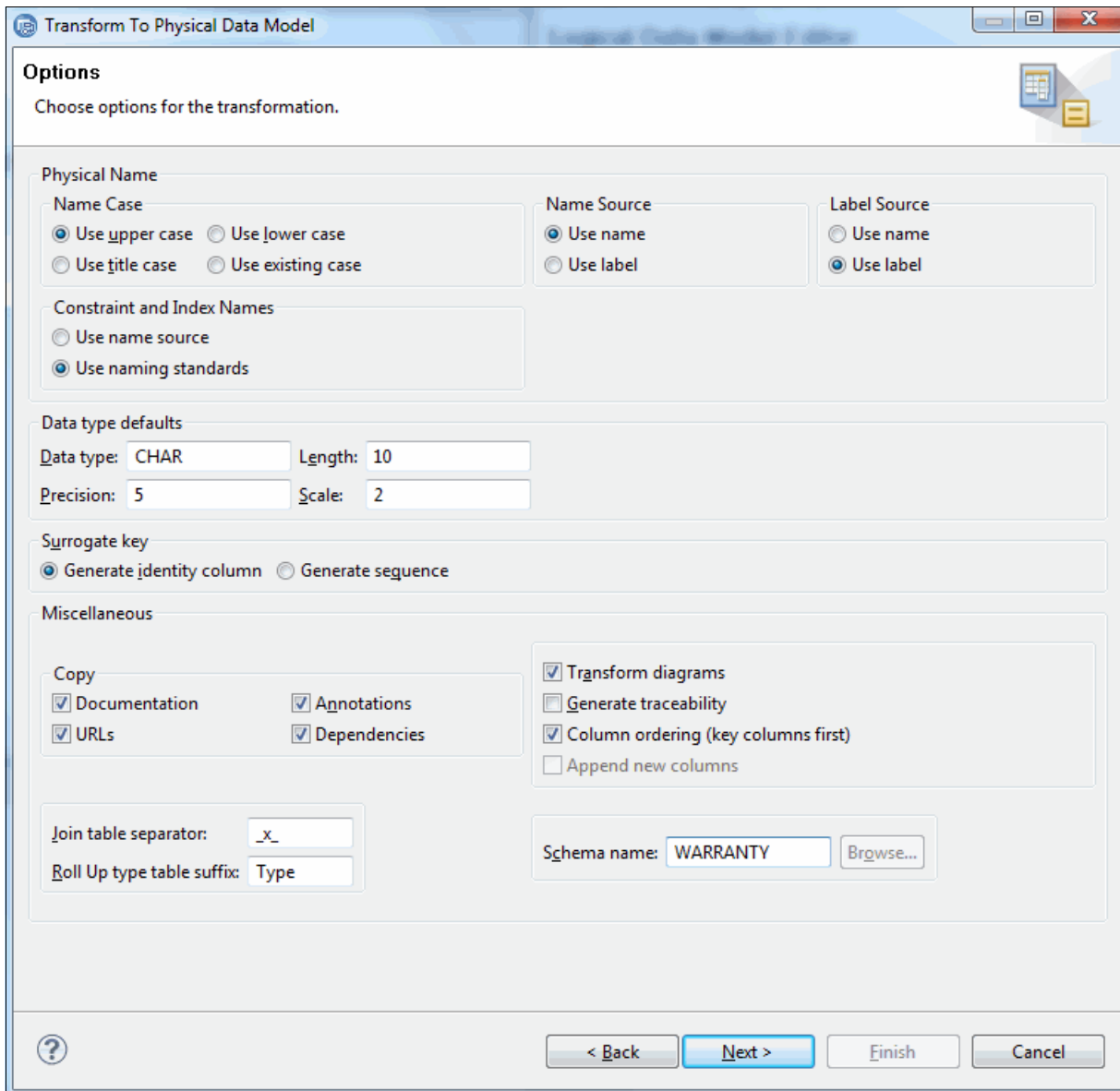


Figure 3-25 Setting the Schema name

5. Once the transformation is complete, a message displays to prompt you to save the generated physical data model by clicking **Finish** (Figure 3-26).

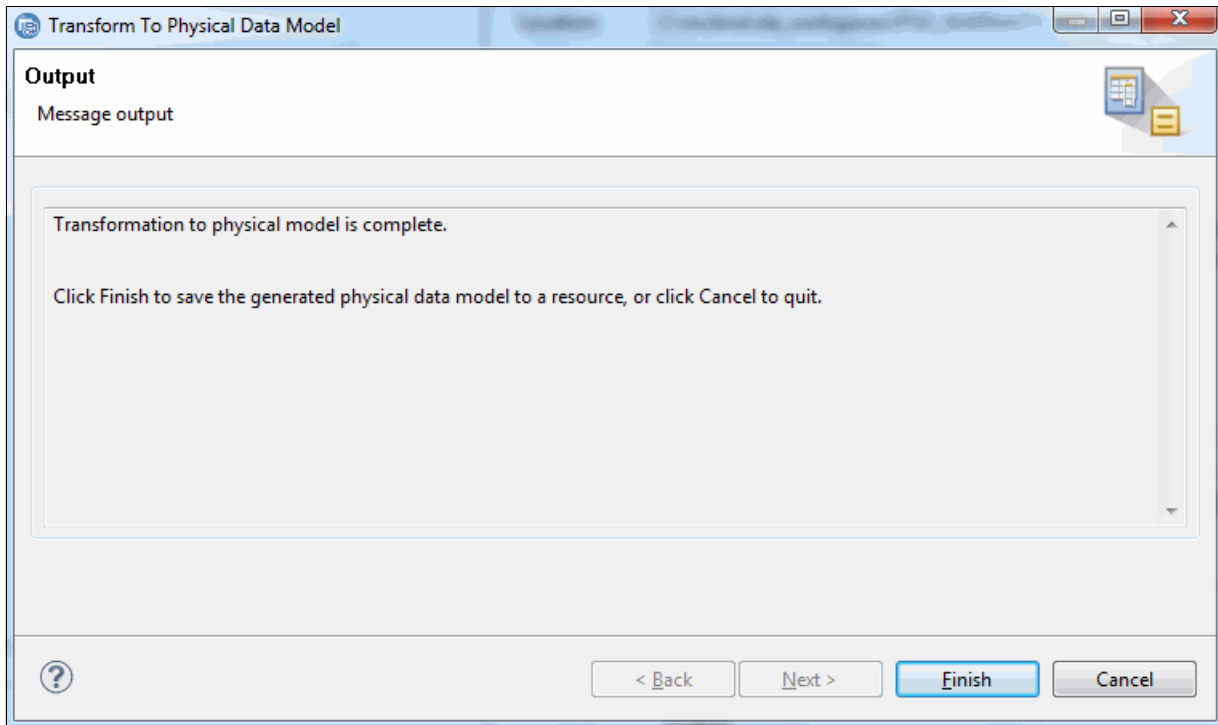


Figure 3-26 Transformation complete

6. In the Data Project Explorer, verify the structure and objects that were created for your physical data model. In addition, from the Properties tab, rename the created database and physical data model diagram objects to something meaningful, such as ITSOWADB and ITSOWARRANTY_Physical_DB_Diagram, respectively. Refer to Figure 3-27.

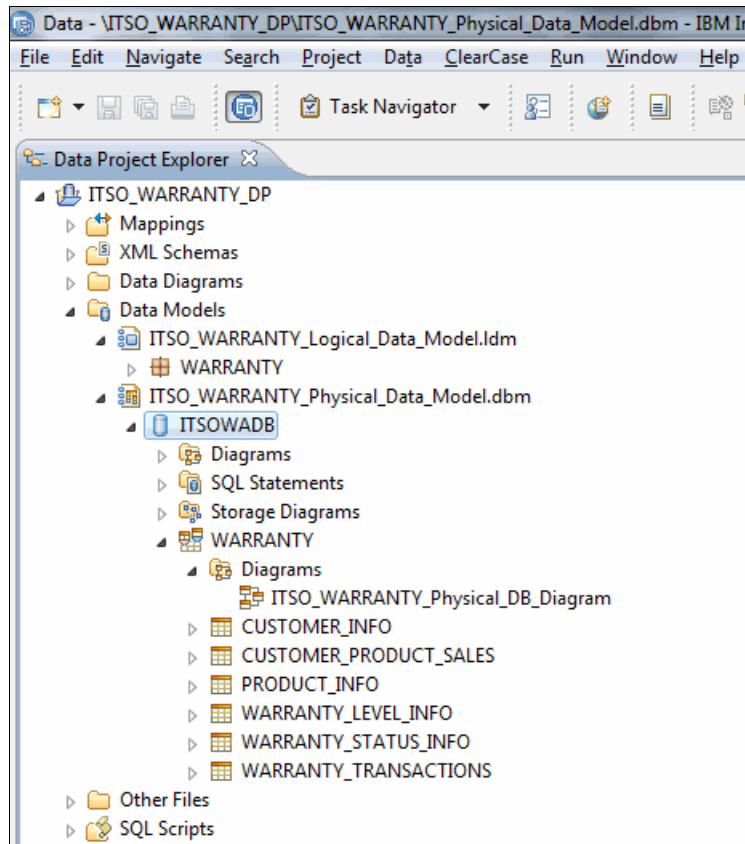


Figure 3-27 Rename the physical data model and the database

7. Next, review your new physical data model diagram to ensure that it represents your data flow properly (Figure 3-28).

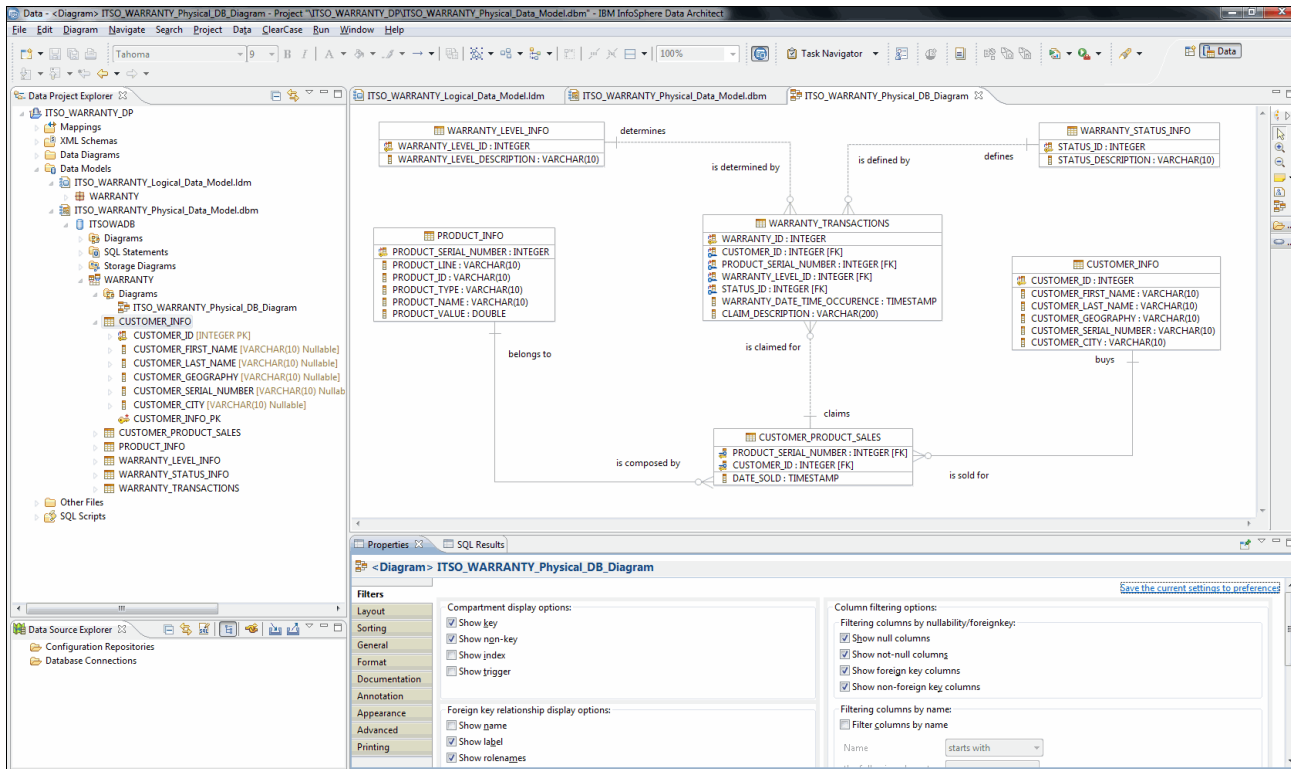


Figure 3-28 WARRANTY physical data model

3.5.3 Generating the DDL

In this example, you produce a DDL script to be able to create the physical database in DB2 with the requirements from the logical and physical data models.

1. Right-click the **ITSOWADB** database and choose **Generate DDL**, as shown in Figure 3-29.

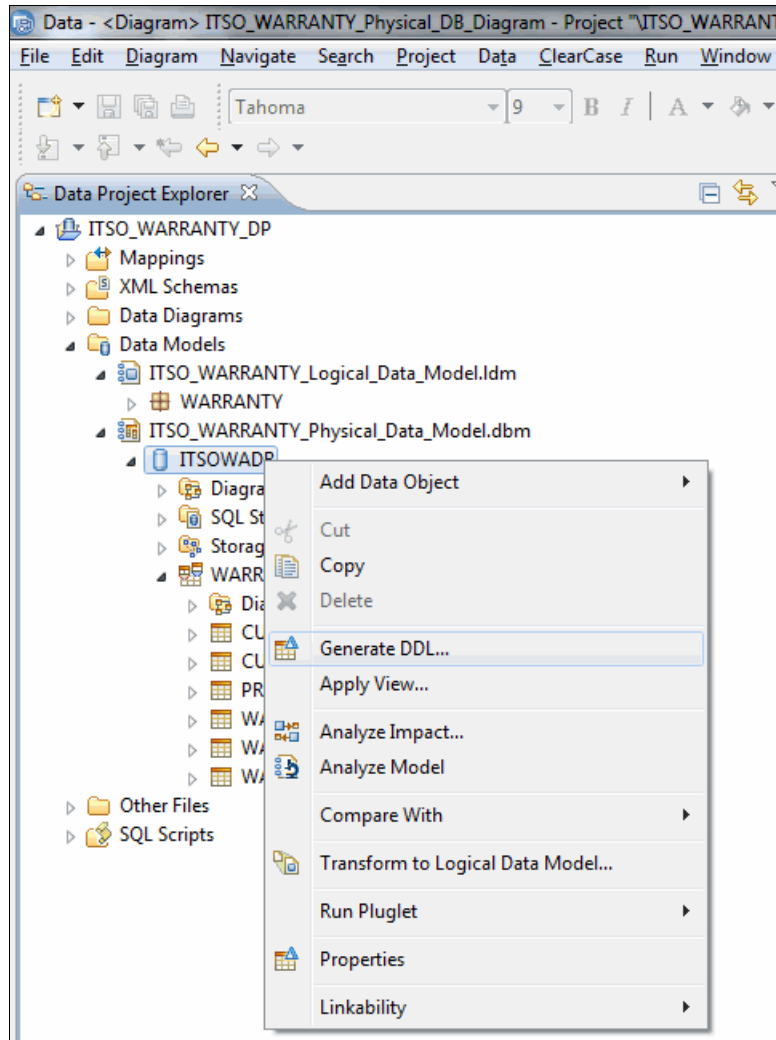


Figure 3-29 Generate DDL

2. Click **Next** to accept the default model elements (that is, options) as shown in Figure 3-30.

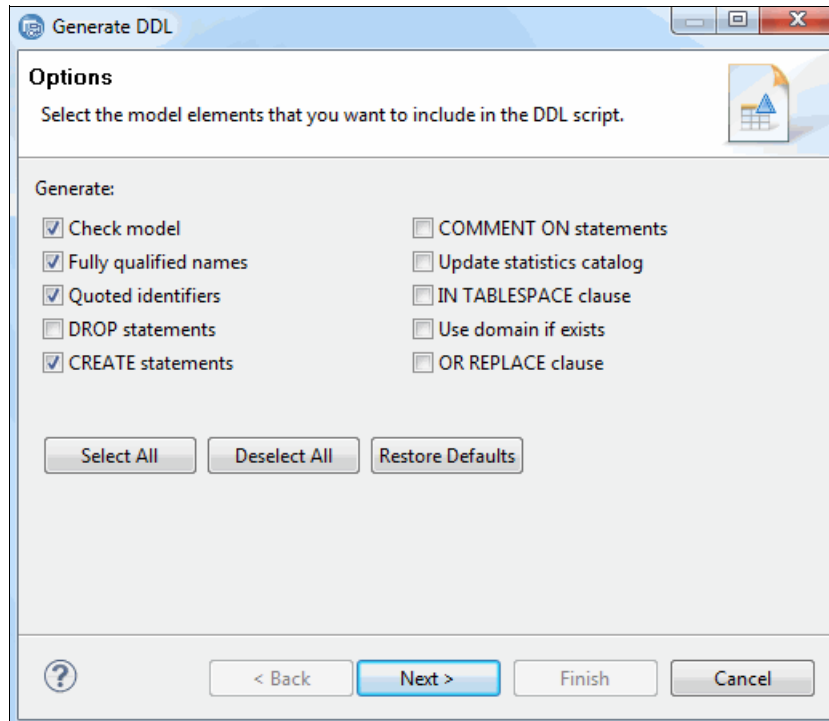


Figure 3-30 Generate DDL options

3. Again, click **Next** to accept the default model objects as shown in Figure 3-31.

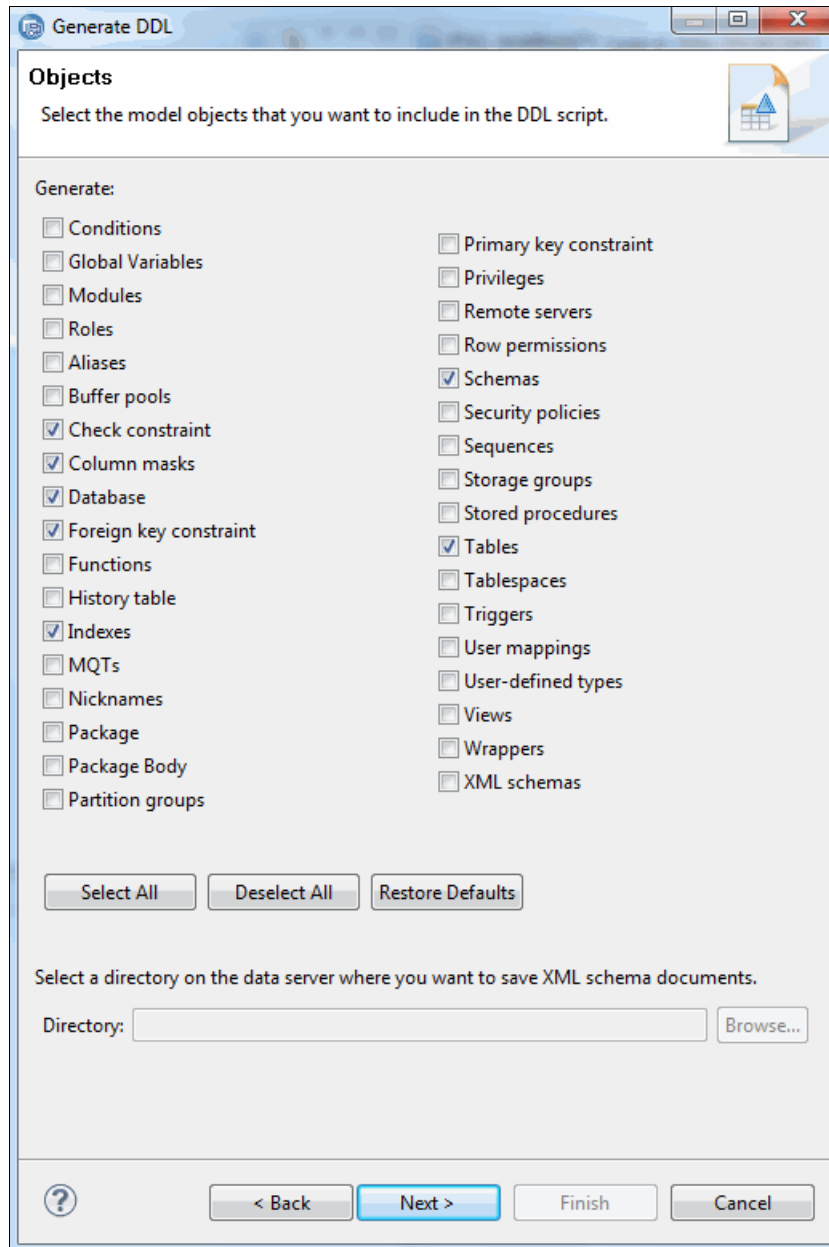


Figure 3-31 Generate DDL

4. Enter a name for your DDL script, such as `WARRANTY_DB.sql`, in the “File name” field, select the **Edit and run DDL in the SQL editor** option, as shown in Figure 3-32, and click **Next** when done.

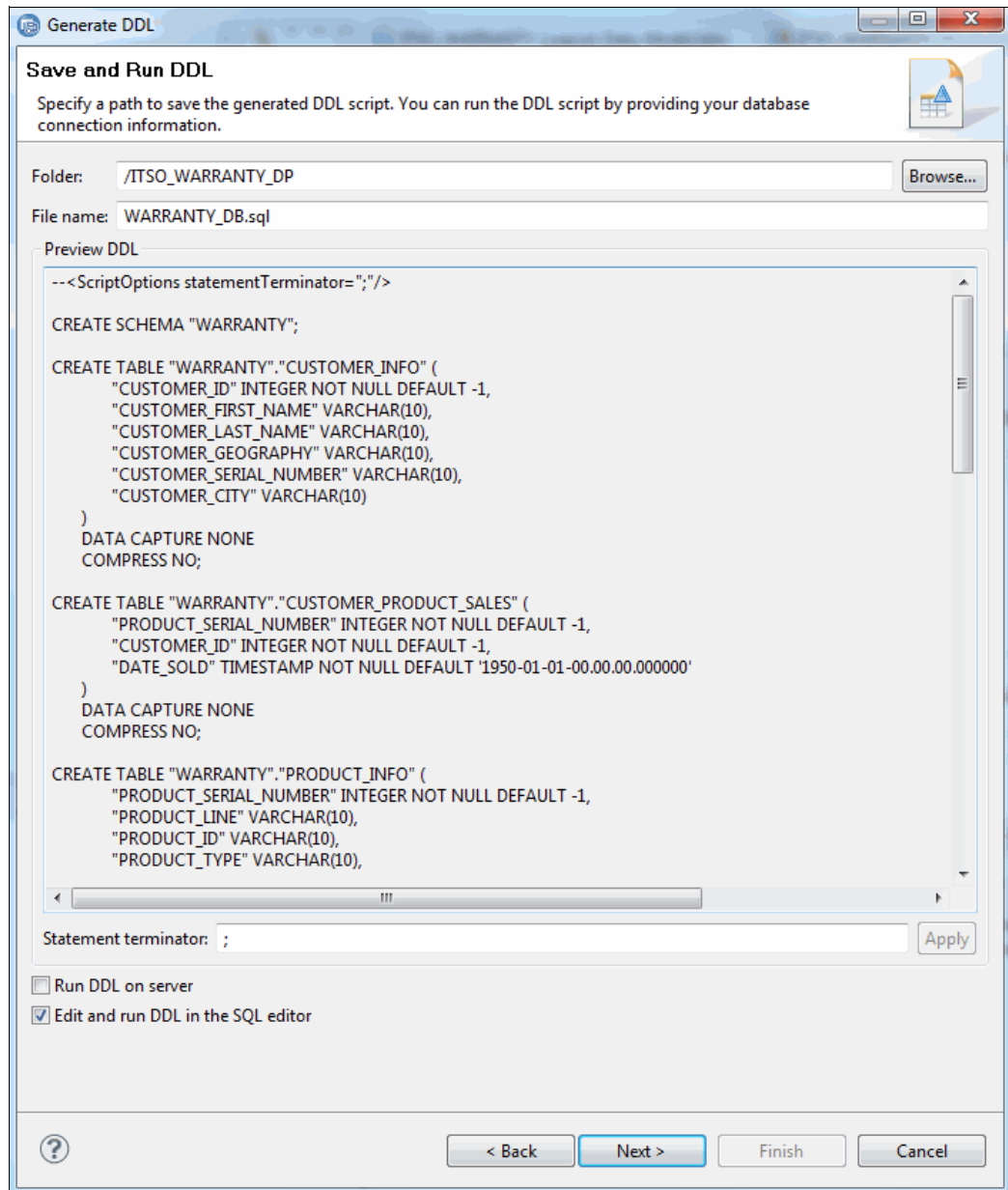


Figure 3-32 Generate DDL

5. Click **Finish** to generate your DDL script.

Your DDL is created under the SQL Scripts Folder in the Data Project Explorer, as shown in Figure 3-33.

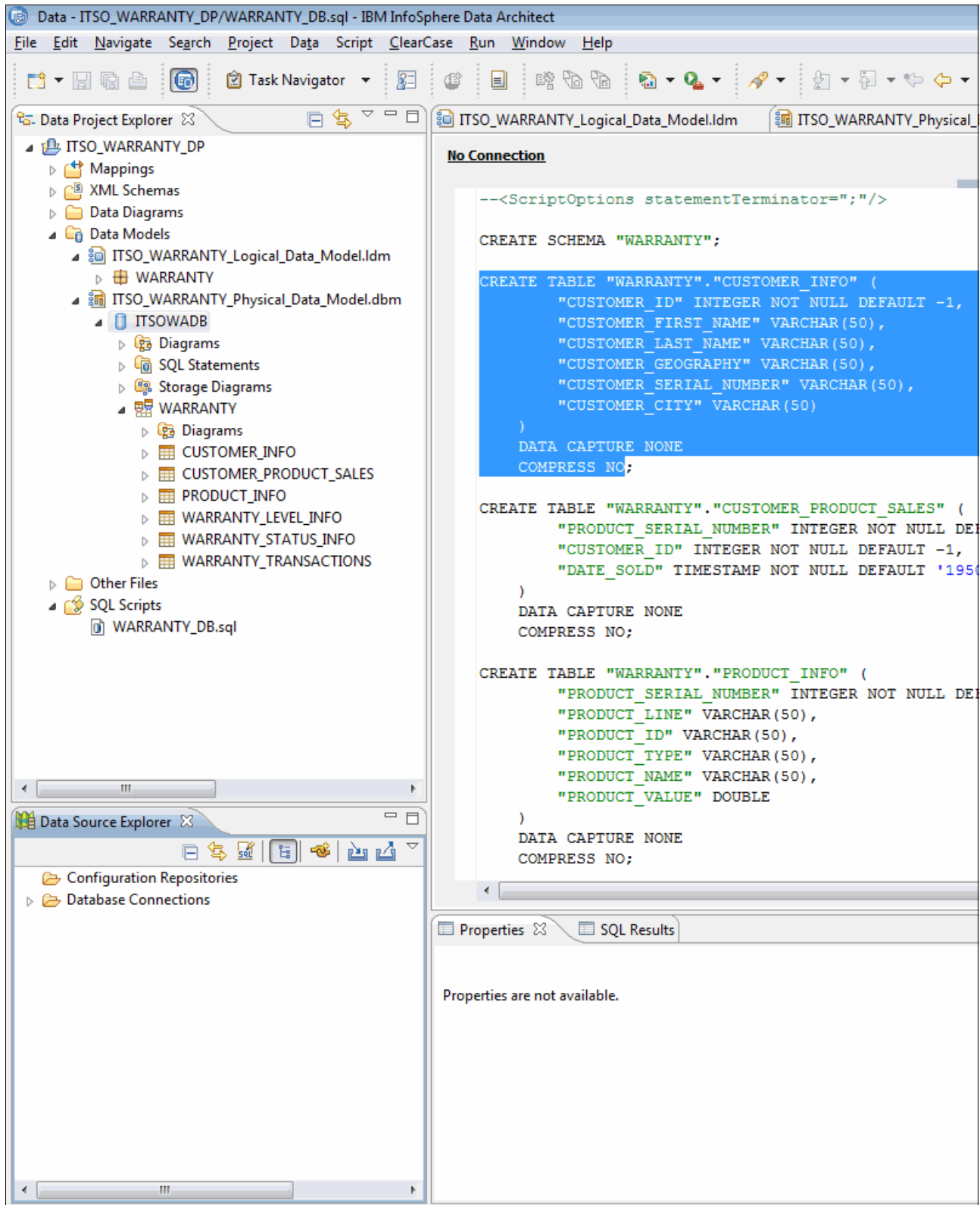


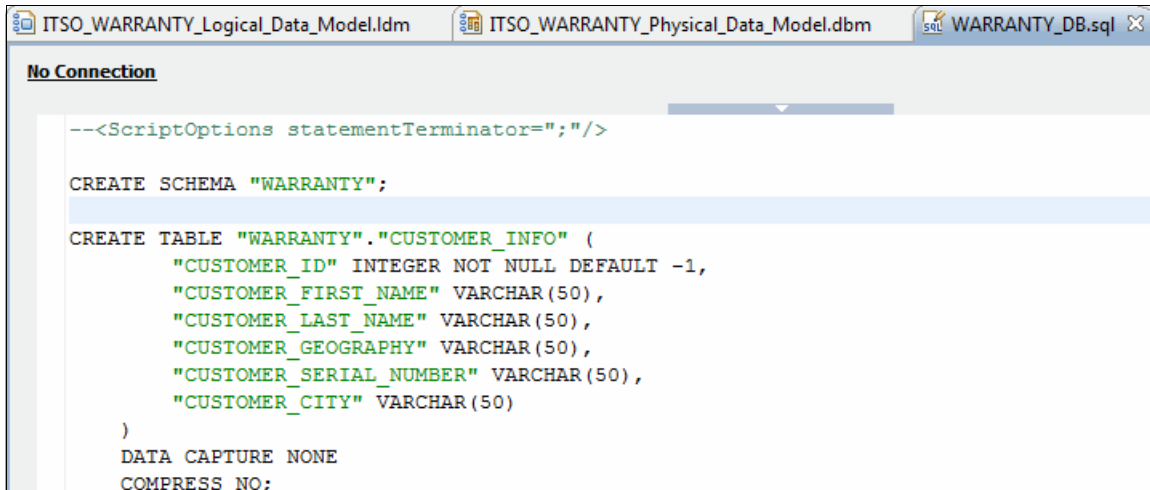
Figure 3-33 Reviewing the DDL

Your script contains and describes all of the database artifacts that the warranty application needs.

3.5.4 Creating the database connection

To create the database connection:

1. Right-click **WARRANTY_DB.sql** under **Data Project Explorer** → **ITSO_WARRANTY_DP** → **SQL Scripts** and choose **Open With** → **SQL and XQuery Editor**.
2. Click the **No Connection** link at the top of the SQL Editor (Figure 3-34).



```
--<ScriptOptions statementTerminator=";" />
CREATE SCHEMA "WARRANTY";
CREATE TABLE "WARRANTY"."CUSTOMER_INFO" (
    "CUSTOMER_ID" INTEGER NOT NULL DEFAULT -1,
    "CUSTOMER_FIRST_NAME" VARCHAR(50),
    "CUSTOMER_LAST_NAME" VARCHAR(50),
    "CUSTOMER_GEOGRAPHY" VARCHAR(50),
    "CUSTOMER_SERIAL_NUMBER" VARCHAR(50),
    "CUSTOMER_CITY" VARCHAR(50)
)
DATA CAPTURE NONE
COMPRESS NO;
```

Figure 3-34 Creating the database connection

3. Click **New** to create a new connection profile (Figure 3-35).

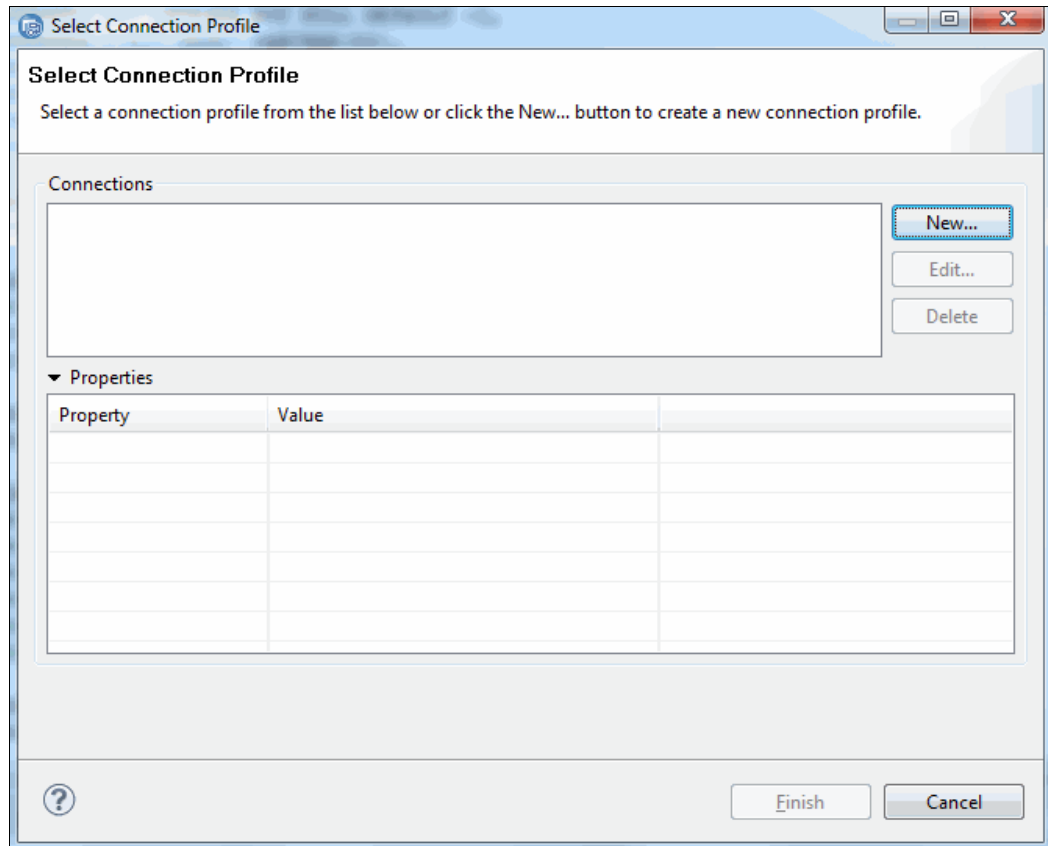


Figure 3-35 Create a new database connection

4. Choose the DB2 for Linux, UNIX, and Windows connection profile (as you can see, there are many profiles to choose from), enter a name, such as ITSOWADB_CON, in the Name field (Figure 3-36), and click **Next**.

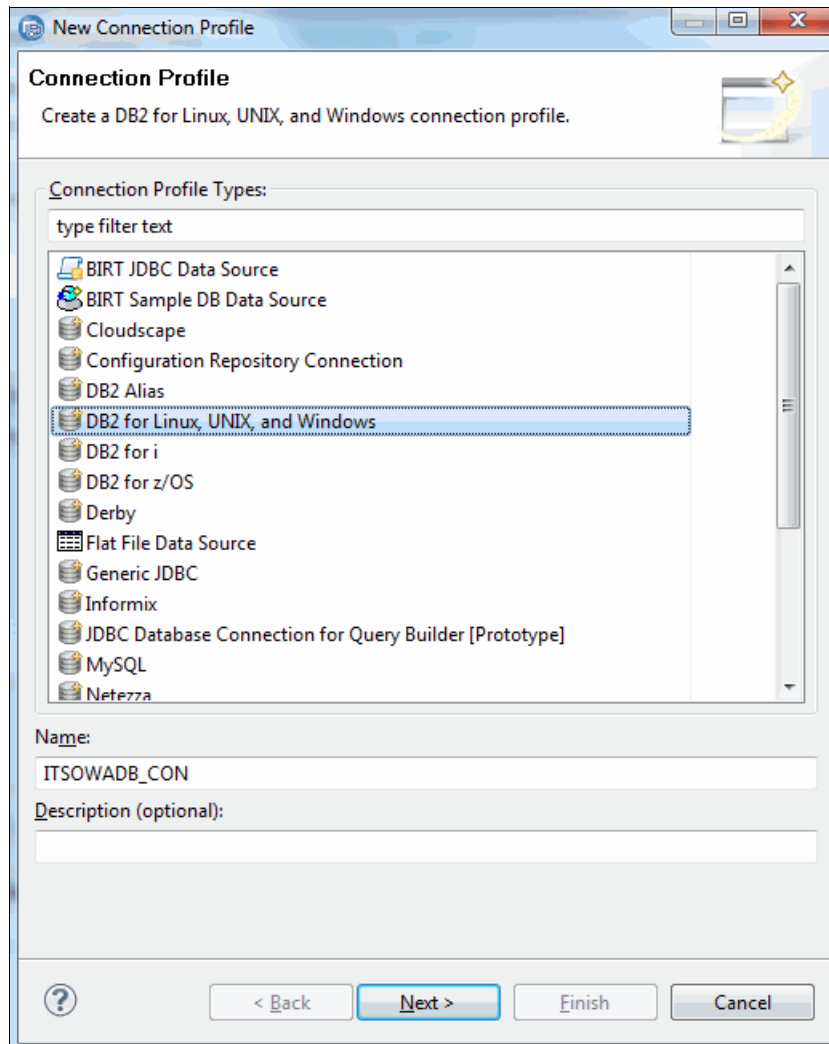


Figure 3-36 New Connection Profile

5. Next, specify the Driver and Connections Details such as database, host, user name, and password.

Note: Obtain this information from your database administrator.

6. if you are replicating this example, enter the details shown in Figure 3-37 on page 60, select the **Save Password** and **Connect when the wizard completes** options, as shown in Figure 3-37, and click **Next**.

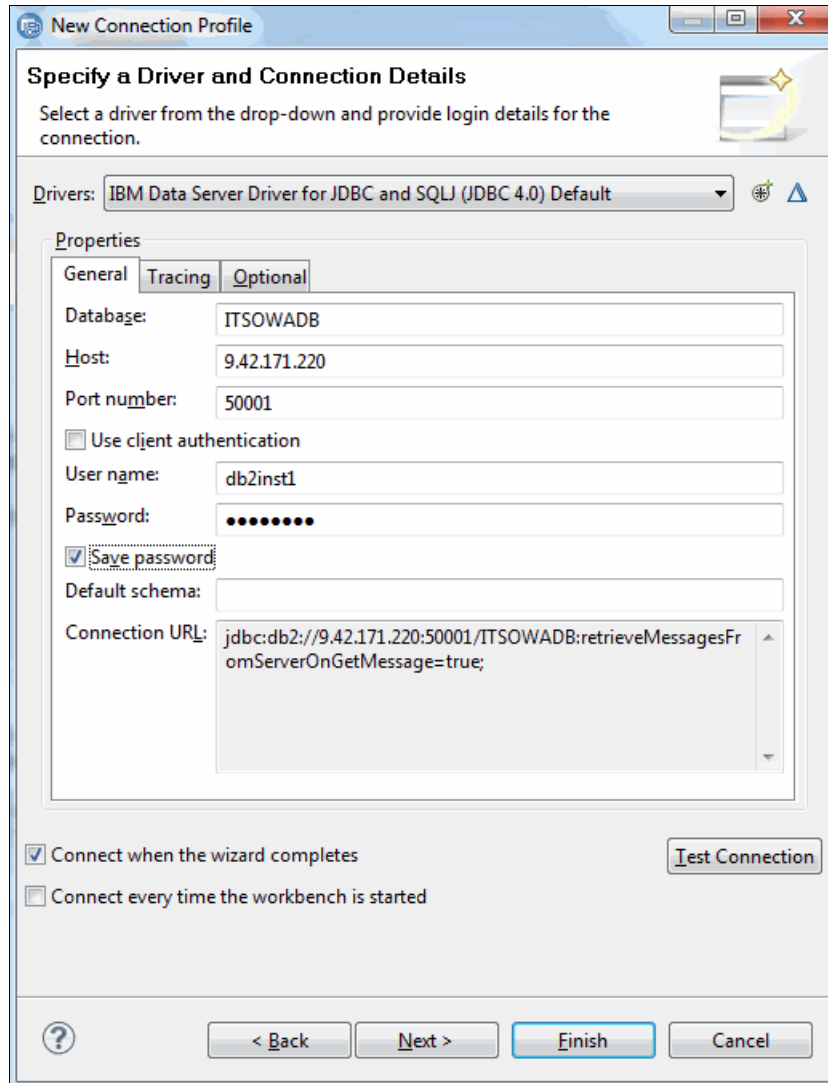


Figure 3-37 Specify Driver and Connections Details

7. Review your Connection Profile and click **Finish** after validating the values (Figure 3-38).

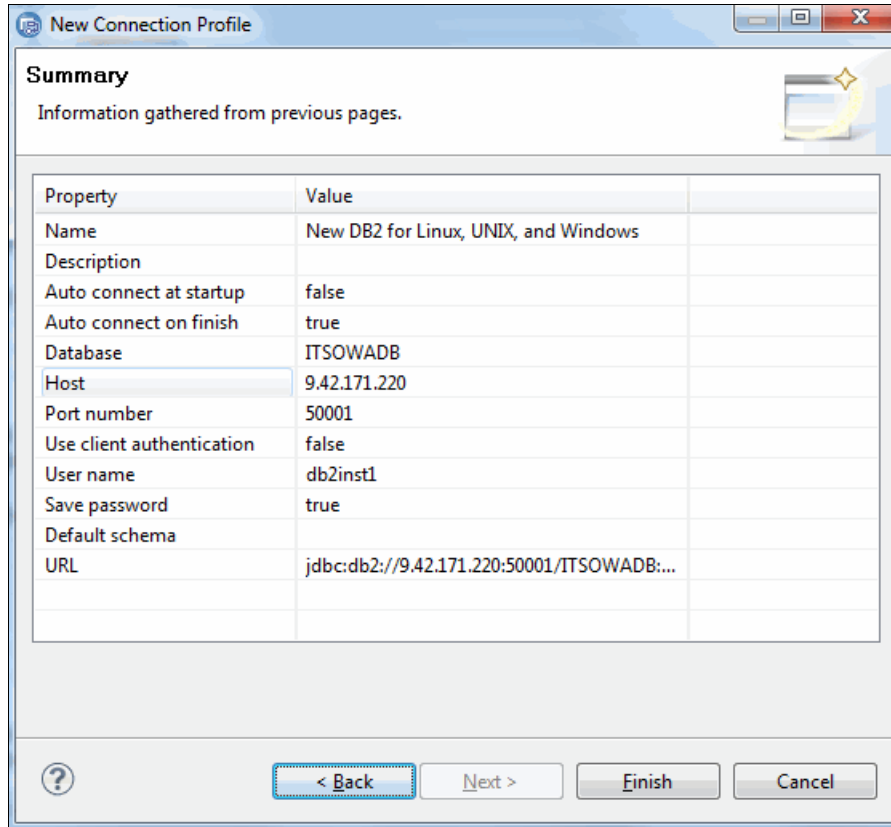


Figure 3-38 Finishing the connection profile

Your new connection profile will appear under the Data Source Explorer section, as shown in Figure 3-39.

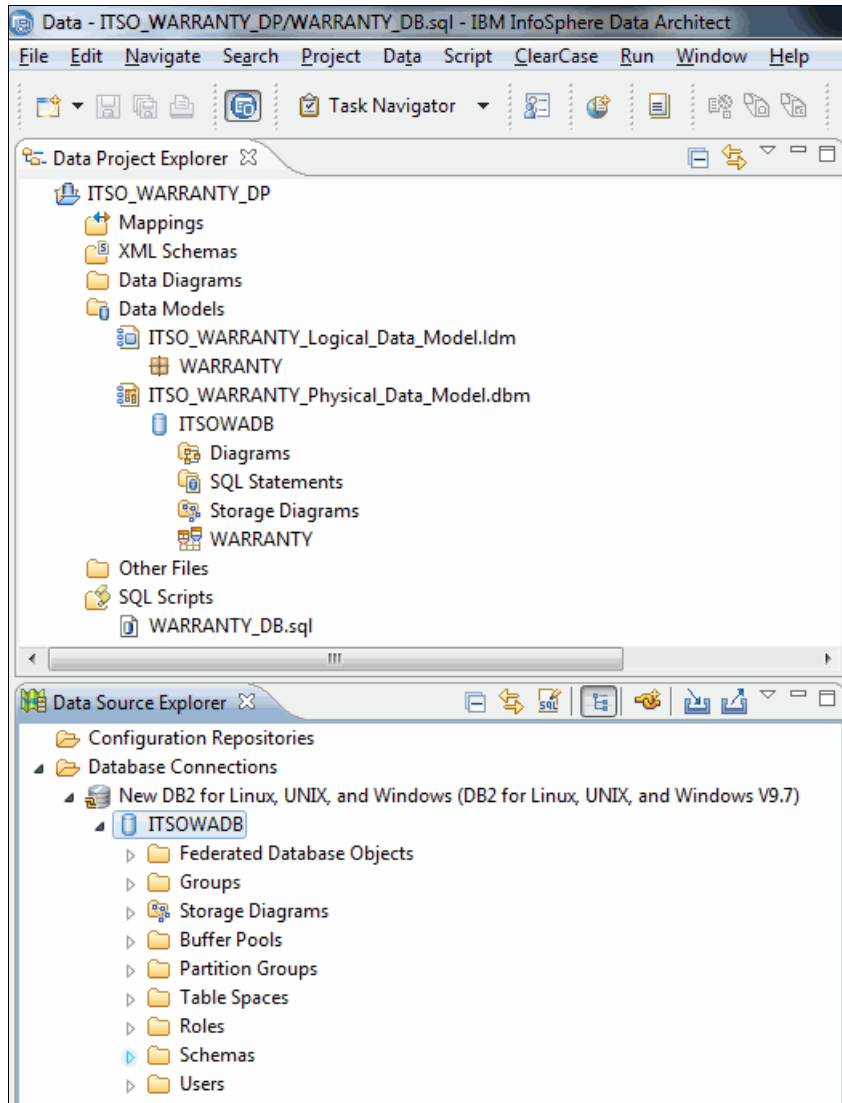


Figure 3-39 Data Source Explorer

3.5.5 Running the DDL and creating the database

Now that you have your DDL script, you are ready to create the database objects and structure for the warranty application. Before running your DDL script, you can review the statements in the SQL Editor Configuration to validate the results.

To run the DDL and create the database:

1. Click **Run SQL** (Figure 3-40) located in the right top side of the SQL Editor.

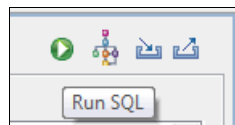


Figure 3-40 Running the DDL in the database server

2. Double-click the **SQL Results** tab and select the Logs (Figure 3-41).

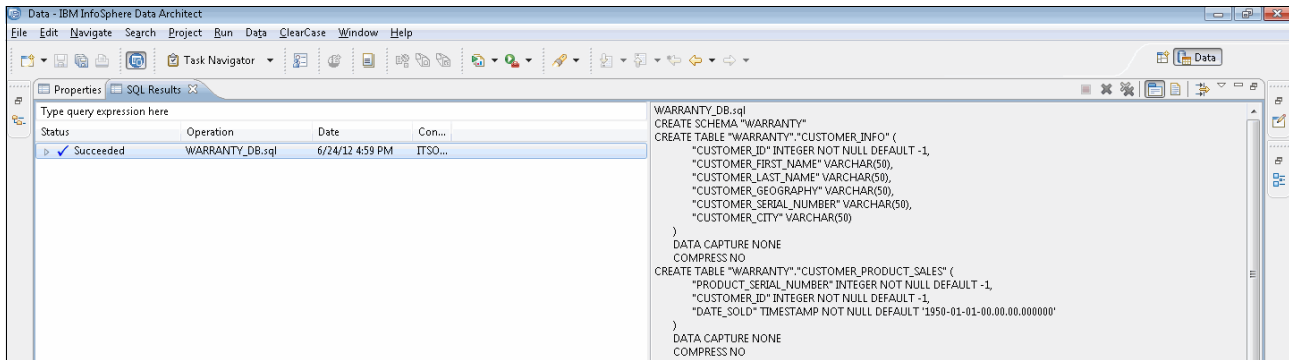


Figure 3-41 SQL results

3. WARRANTY_DB.sql creates the WARRANTY schema, tables, columns, and all needed objects in the database successfully. Scroll down the list to see the total execution time (Figure 3-42).

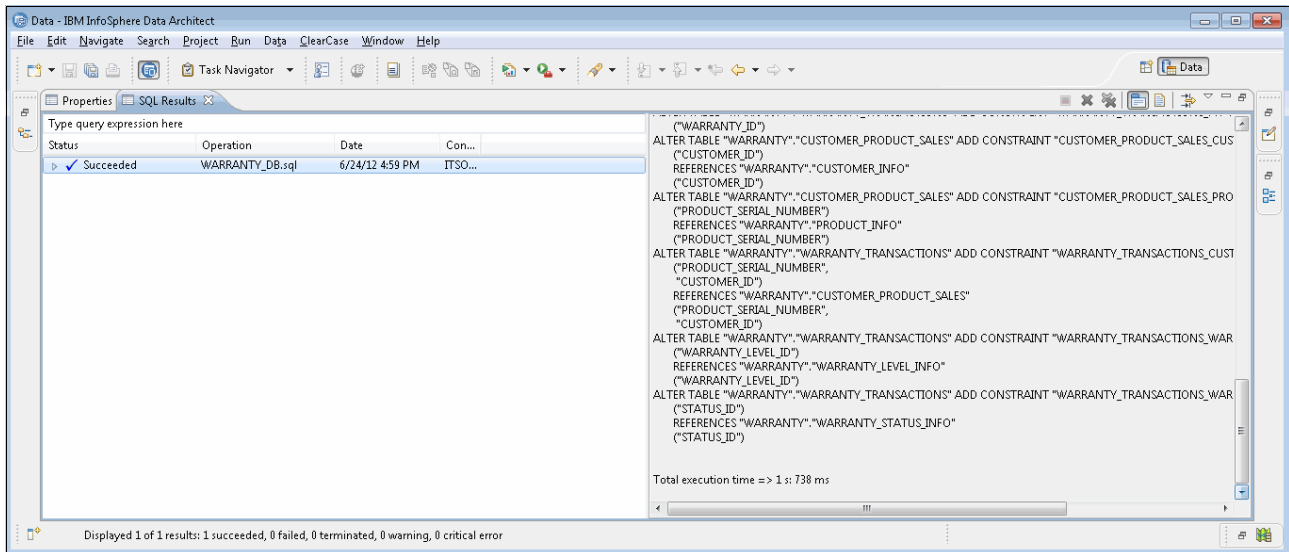


Figure 3-42 Total execution time

4. In the Data Source Explorer, right-click **Database Connections** and then **ITSOWADB_CON** → **ITSOWADB** → **Schemas**, and then select **Refresh** (Figure 3-43).

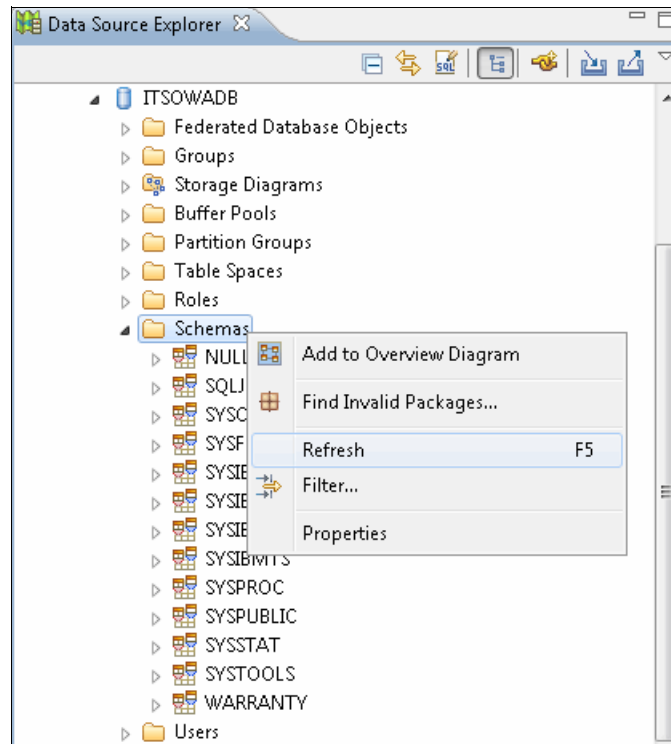


Figure 3-43 Refreshing the database schemas

- The WARRANTY schema will show under the Schemas folder in the Data Source Explorer. (Figure 3-44).

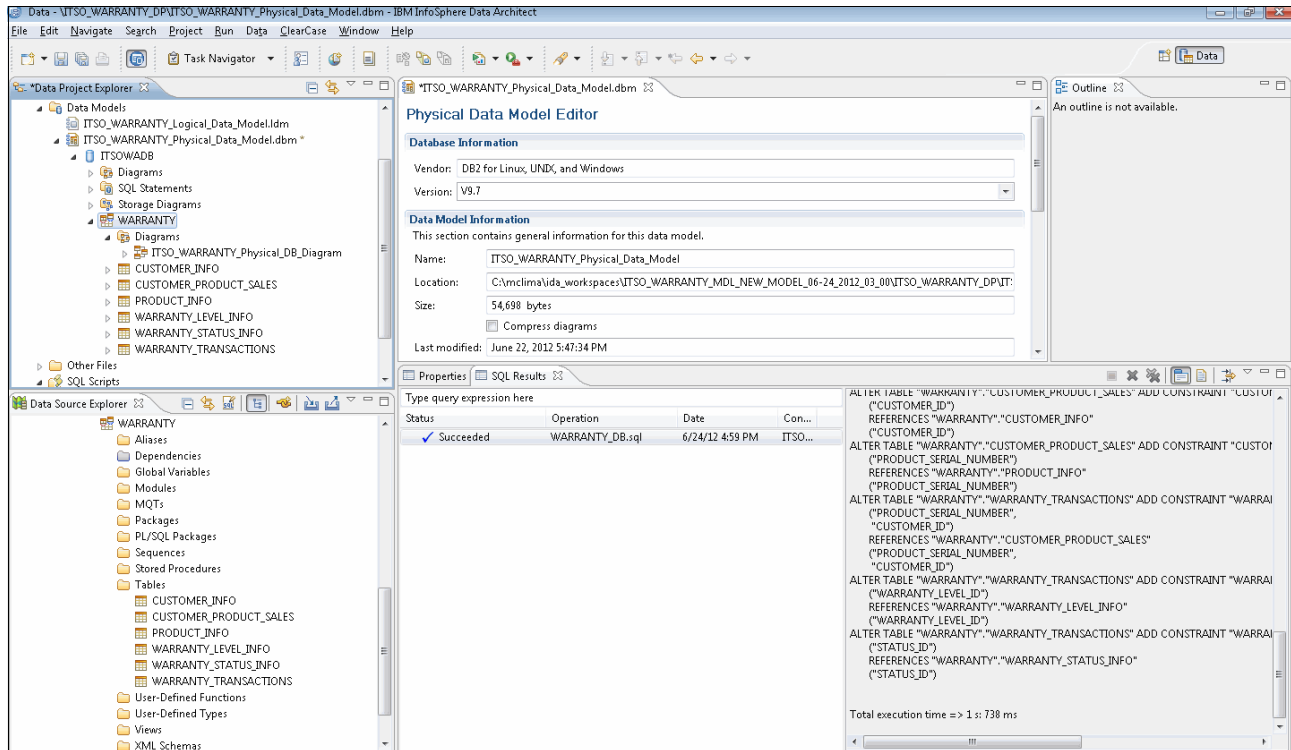


Figure 3-44 The Data Source Explorer and the WARRANTY schema

3.6 Importing data into the WARRANTY database

Now that your data models have been built and the entities, attributes, and relationships are defined and added, you are ready to import data into the WARRANTY database. You can find the text files referenced in this section in the additional materials shipped with this book. See Appendix A, “Additional material” on page 285.

1. In the Data Source Explorer, select **ITSOWADB_CON** → **ITSOWADB** → **Schemas** → **WARRANTY** → **Tables** (Figure 3-45).

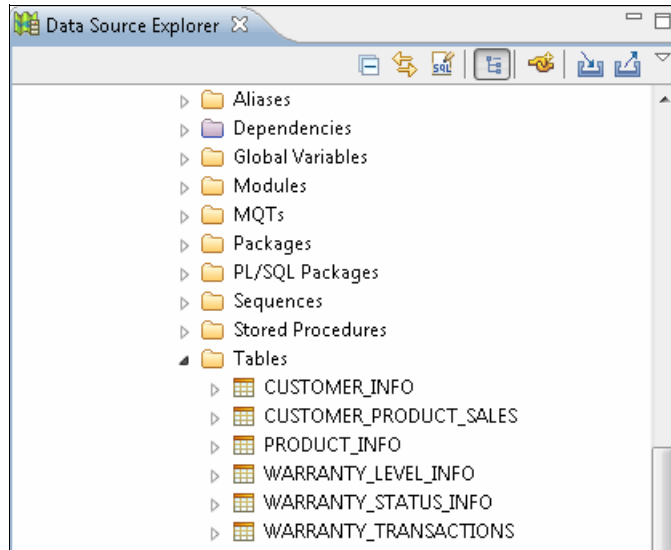


Figure 3-45 Importing data into the WARRANTY database

2. Right-click the CUSTOMER_INFO table and select **Data** → **Load**.
3. Next, select the CUSTOMER_INFO_DATA.txt data file as the Input File by using the Browse button, and then select **Comma** as column delimiter and **None** as the Character string delimiter (Figure 3-46); click **Finish** when done.

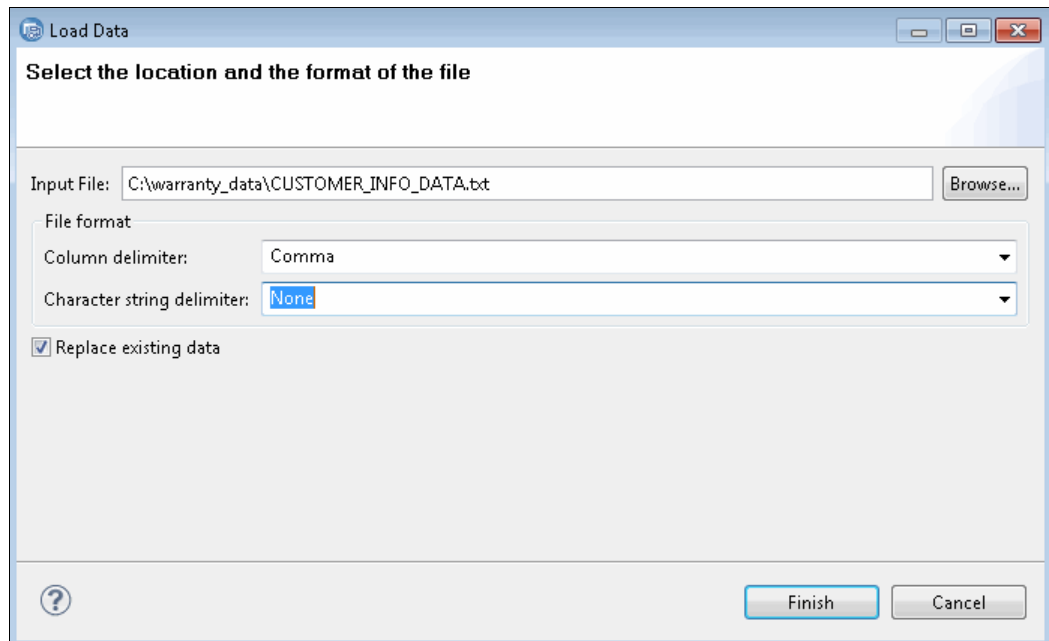


Figure 3-46 Selecting the CUSTOMER_INFO_DATA.txt data file

4. Review the SQL Results log (Figure 3-47 on page 67).

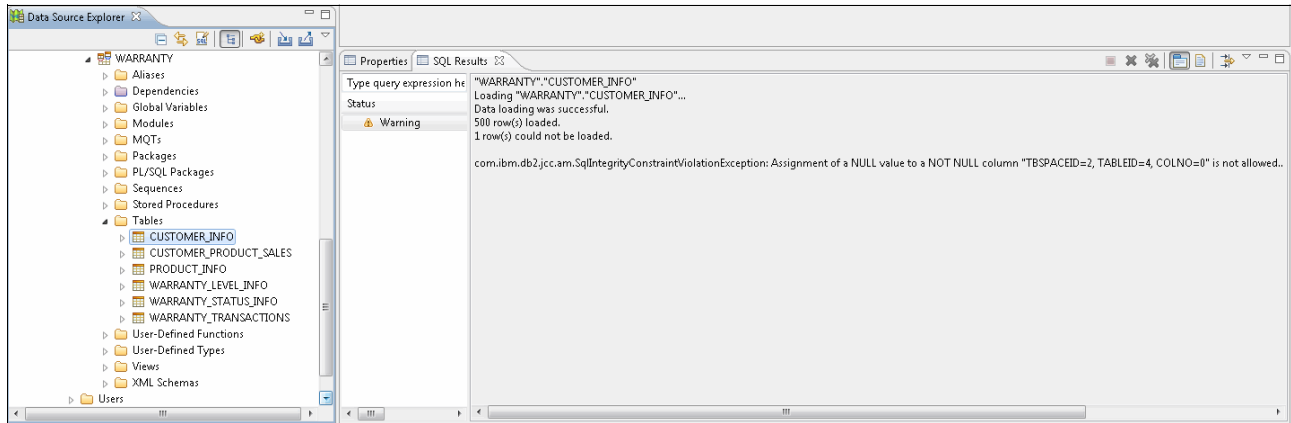


Figure 3-47 The SQL Results log

Tip: You can ignore the warning shown in the SQL Results log, because it refers to the table column labels that are in the first line of the imported data file.

5. Right-click the CUSTOMER_INFO table and select **Data** → **Return All Rows** to view the data in the SQL Results tab (Figure 3-48).

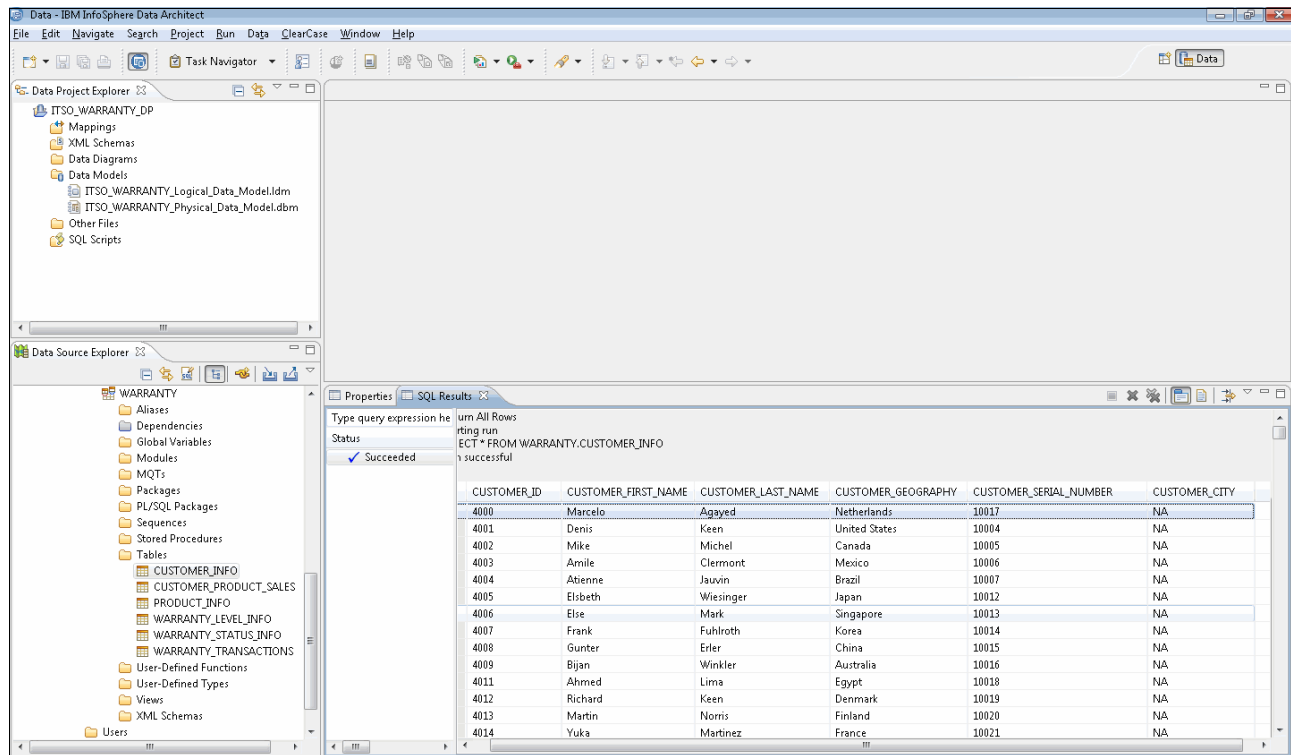


Figure 3-48 Loading data into the WARRANTY database

6. Repeat these steps and import the data files listed in Table 3-1.

Table 3-1 Import data files into the WARRANTY database

Table name	Import data files
PRODUCT_INFO	PRODUCT_INFO_DATA.txt

Table name	Import data files
CUSTOMER_PRODUCT_SALES_INFO	CUSTOMER_PRODUCT_SALES_INFO_DATA.txt
WARRANTY_LEVEL_INFO	WARRANTY_LEVEL_INFO_DATA.txt
WARRANTY_STATUS_INFO	WARRANTY_STATUS_INFO_DATA.txt
WARRANTY_TRANSACTIONS	WARRANTY_TRANSACTIONS_DATA.txt

Tip: Our database was designed with referential integrity. To avoid errors during a load, follow the order presented in Table 3-1.

3.7 Exporting the WARRANTY data model to Cognos

Our scenario uses IBM Cognos Business Intelligence as the reporting and analysis front-end system for business users. You can use IBM InfoSphere Data Architect to export your data model to create a Cognos model.

To export the WARRANTY model to Cognos:

1. Click **File** → **Export** in your toolbar, select **Data Model Export Wizard** from the Data folder (Figure 3-49) and then click **Next**.

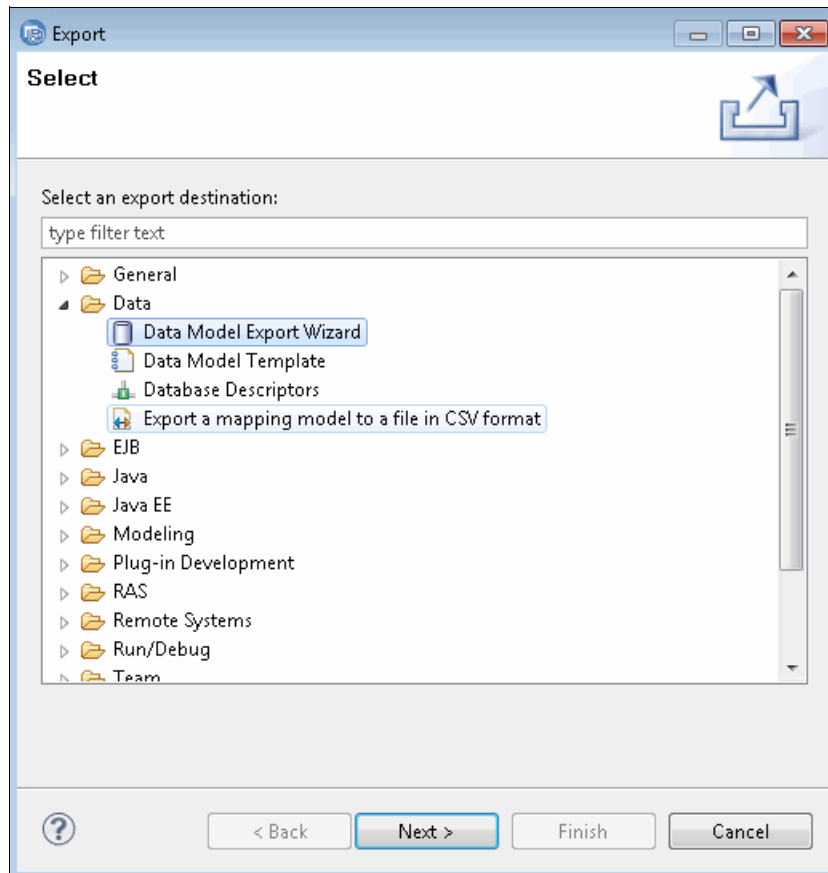


Figure 3-49 Exporting the WARRANTY data model to Cognos

2. In the Data Model Export dialog, change the model format to **IBM Cognos BI Reporting - Framework Manager**, browse your file system to select the Physical Warranty Data Model and conform where to export the Cognos data model (Figure 3-50), and click **Next**.

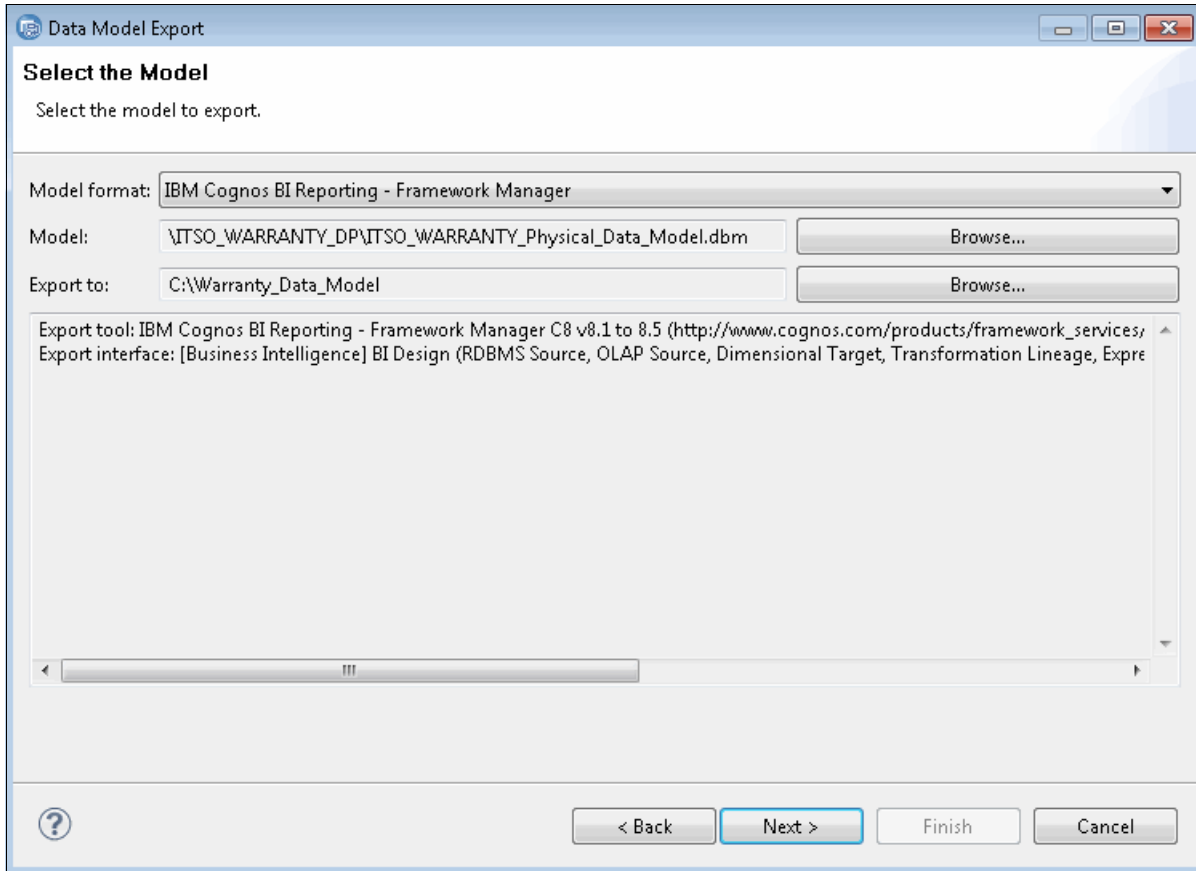


Figure 3-50 Exporting to IBM Cognos Framework Manager

3. Change the Cognos FM Version value to **Cognos 10.1 - XML V60.2**, and then scroll down to the **Datasource name** entry, as shown in Figure 3-51.

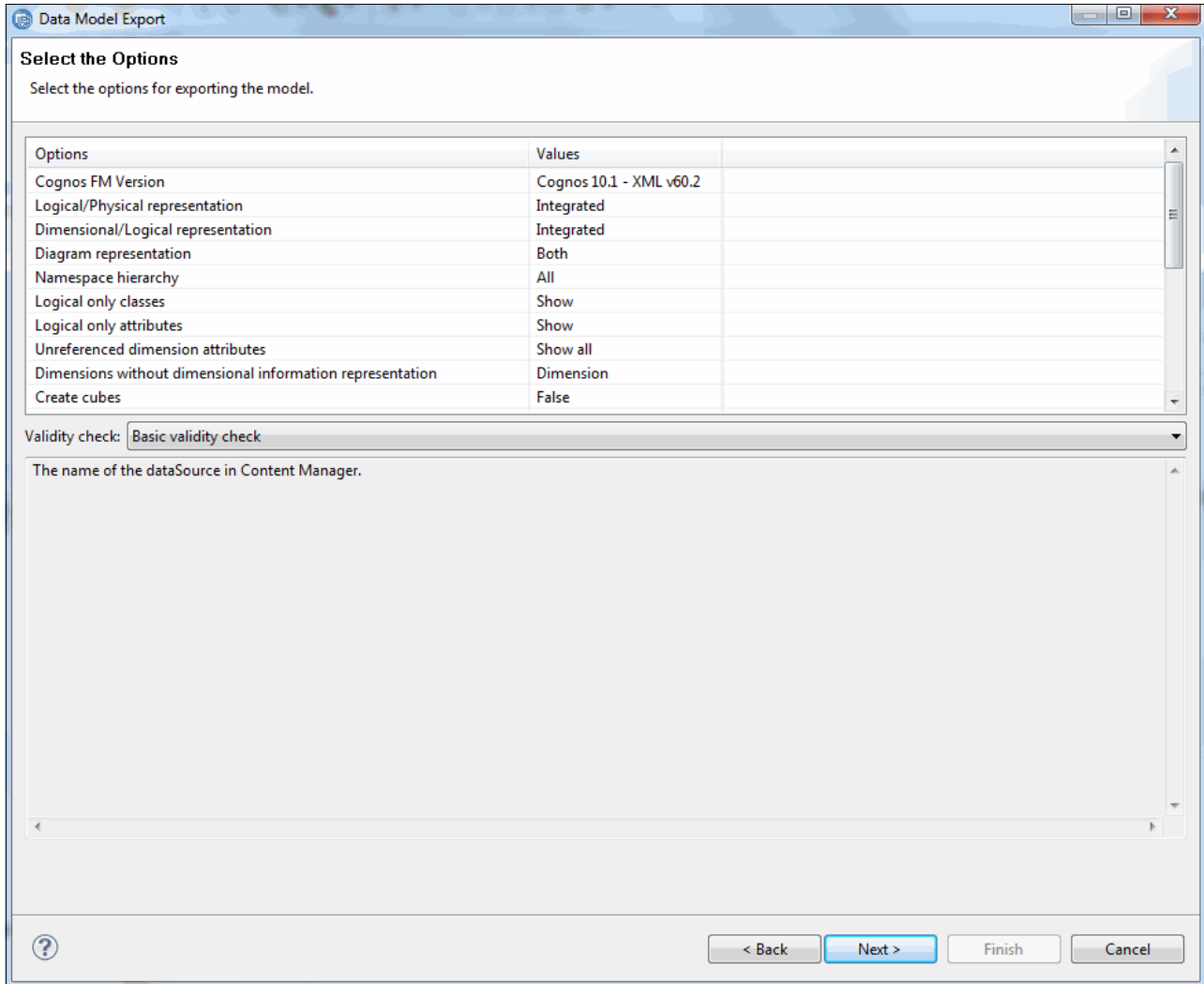


Figure 3-51 Change the version number

4. Enter a name, such as Warranty claims database, in the “Datasource name” field (Figure 3-52).

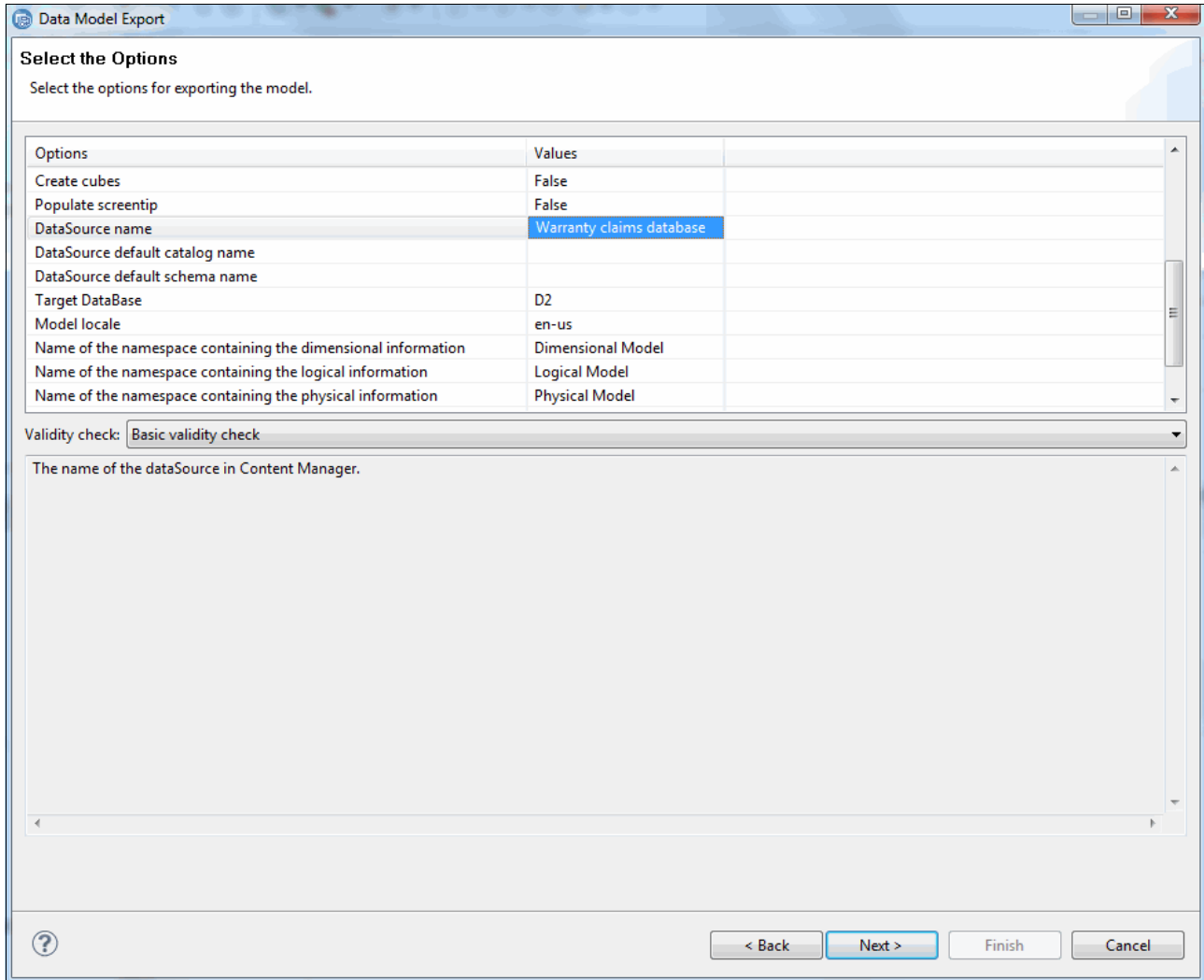


Figure 3-52 Provide the data source name

5. Change the **Create cubes** option to **False**, set the Target DataBase to **D2** (Figure 3-53), and click **Next**.

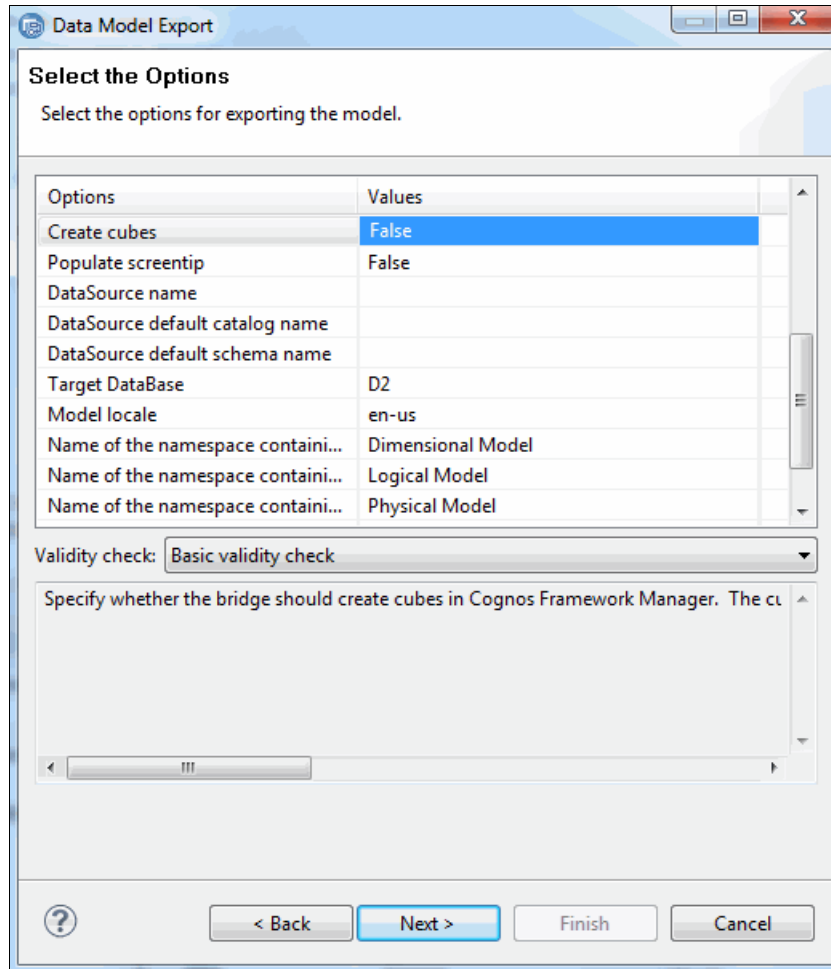


Figure 3-53 Select the Options

6. Review the export summary (Figure 3-54) and click **Finish**.

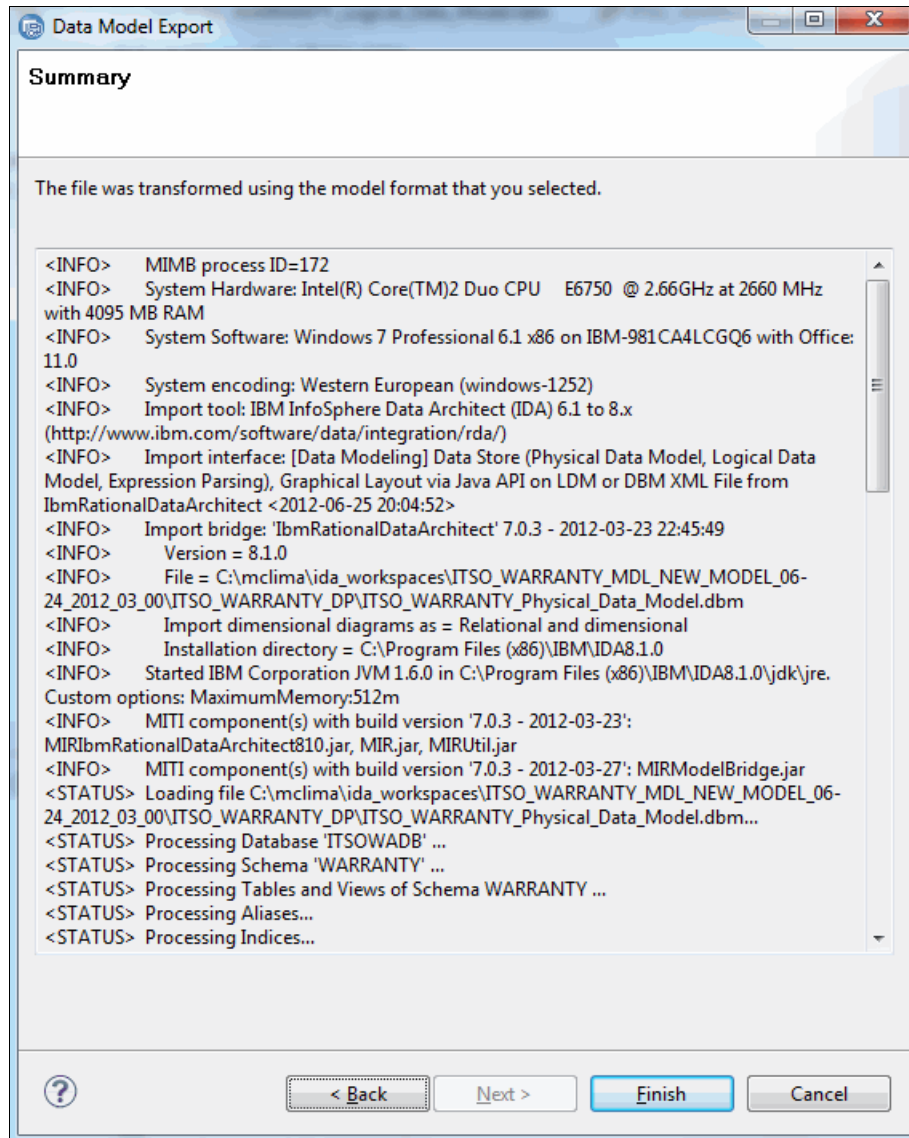



Figure 3-54 Summary: Exporting IBM InfoSphere Data Architect Model to Cognos



Decision management with WebSphere Operational Decision Management

In this chapter we provide an overview describing decision management and define how it complements business process management. In addition, there is detailed guidance for implementing WebSphere Operational Decision Management V8 within the context of an IBM Business Process Manager V8 based solution.

This chapter is organized in three sections:

- ▶ Rule-based decisions are described in 4.4, “Overview of business rules” on page 84.
- ▶ Event-based decisions are described in 4.5, “Overview of Event Rules” on page 92.
- ▶ Section 4.6, “Business management of rules” on page 105 explains how an organization can enable their business users to manage rules effectively.

4.1 Business drivers for decision management

Every business application, whether driven by a business process as described in this book, or some alternative, must implement some form of business rules. Traditionally, business rules **have** been implemented in a programming language such as Java or COBOL. Programming the business rules into an application leads to a number of concerns and problems that arise in development and maintenance of the application. For example:

- ▶ The rules written in a programming language are not accessible in a form that is understandable by those who define those rules, namely the business user.
- ▶ Since some applications implement the same policies, having tightly embedded rules within one application makes it harder for the others to access these rules.
- ▶ At times, there may be no standard for business rules, which causes applications to implement them inconsistently.
- ▶ Since programming rules require resources to develop, test and analyze them, this process can become costly in real dollars and time to implement.
- ▶ Auditing rule execution to ensure it supports business requirements is often difficult to produce meaningful assessments.
- ▶ Simulating new rules is often time consuming, which impacts the ability of the business to react quickly.

A decision management solution resolves these issues by providing a single source of truth for an organization's business policy, which can enable it to streamline the development, testing, simulation, and auditing of its business policies. A decision management system should be constructed so that non-technical business users can easily view, maintain and test rules, often with little or no assistance from technical resources.

In this chapter we use IBM WebSphere Operational Decision Management as the decision management solution (Figure 4-1 on page 77) to define the rules for the warranty application.

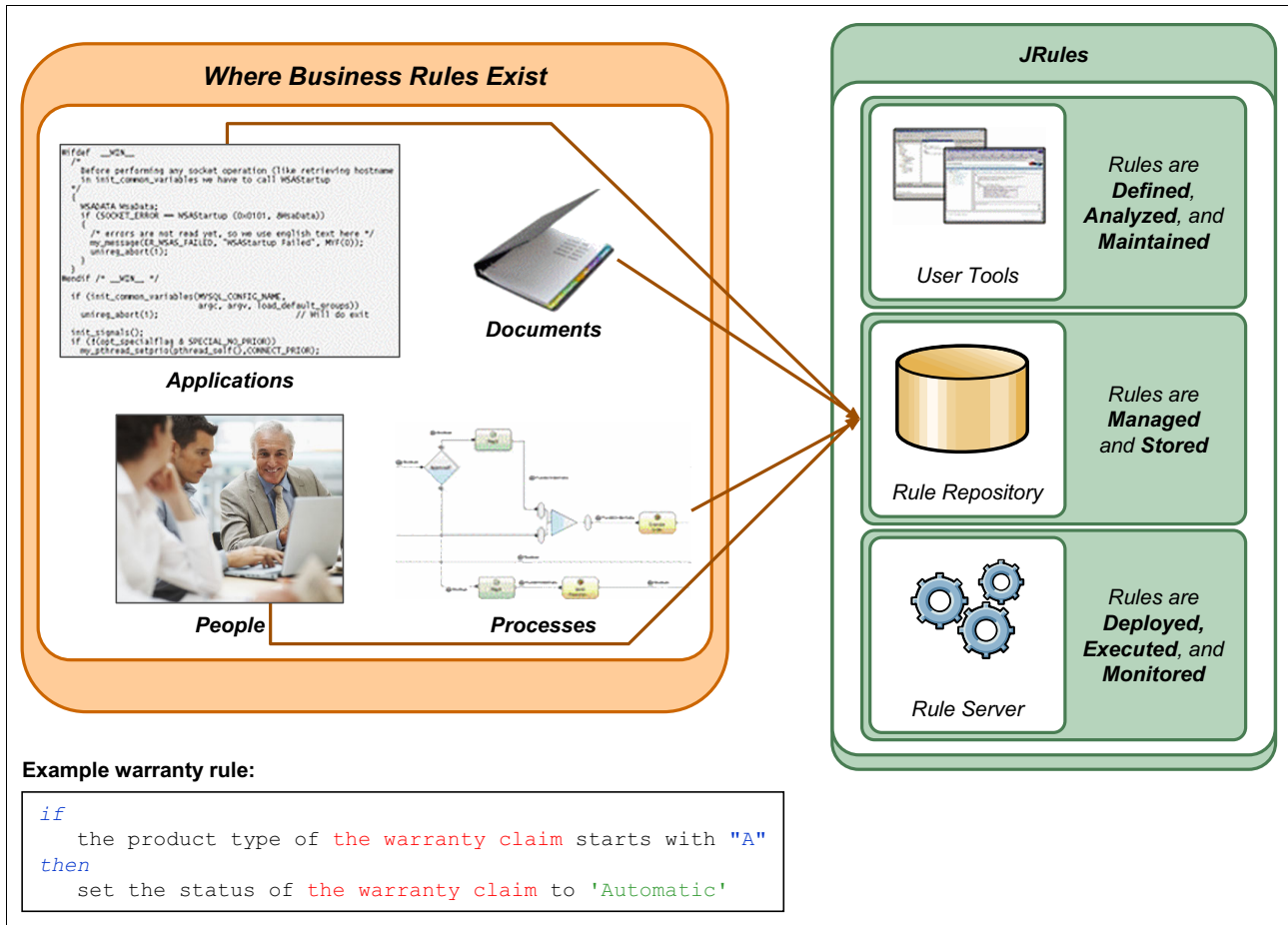


Figure 4-1 Business rules inside WebSphere Operational Decision Management

4.1.1 The difference between business rules and event rules

Business rules and event rules: to place these concepts in a business context, we describe how a company that processes warranty claims can manage its process using IBM WebSphere Operational Decision Management.

The following rule might be implemented in a warranty system:

```

if
the value of the warranty claim is greater than $1,000
then
set manager approval to mandatory

```

As part of the same system, it may also contain a rule like this:

```

if
the number of warranty claims of the customer within the last month is
greater than 4
then
send a request to the call center to contact the customer

```

From a business user's perspective, both of the above are "rules". These examples reflect a business policy that he or she may want their warranty system to implement.

Our first example is what we call a “business rule,” while the second we refer to as an “event rule”. The distinguishing difference between the two is that the first has no reference to time, while the second does (for example “within the last month”).

From a technical perspective, this small difference will impact how each rule functions from an architectural perspective:

A business rule application module:

- ▶ Receives data synchronously.
- ▶ Responds synchronously.
- ▶ Is stateless.

An event rule application module:

- ▶ Is called asynchronously.
- ▶ May trigger an asynchronous response.
- ▶ Is stateful if the rule correlates the current event with any previous events.

These technical differences are covered in the following sections.

4.2 Decision management with BPM

Decision management does not and cannot function in isolation: the decision management runtime is a back-end service that must interact with other applications to add value. While it is possible to plug decision management into any application, it is best applied in business process applications.

In many enterprises, business process management and decision management are complementary capabilities. Simply put:

- ▶ A business process tells you what to do.
- ▶ Decision management tells you how to do it.

Sometimes organizations attempt to map business processes without using decision management. If the decisions that drive these processes are simple and infrequent, and if those decisions do not interact with other systems, then it is possible to embed your decision logic within a process. However, implementing complex decisions within a business process without some form of decision management can lead your processes to become large, complex and difficult to understand and maintain, as shown in Figure 4-2 on page 79.

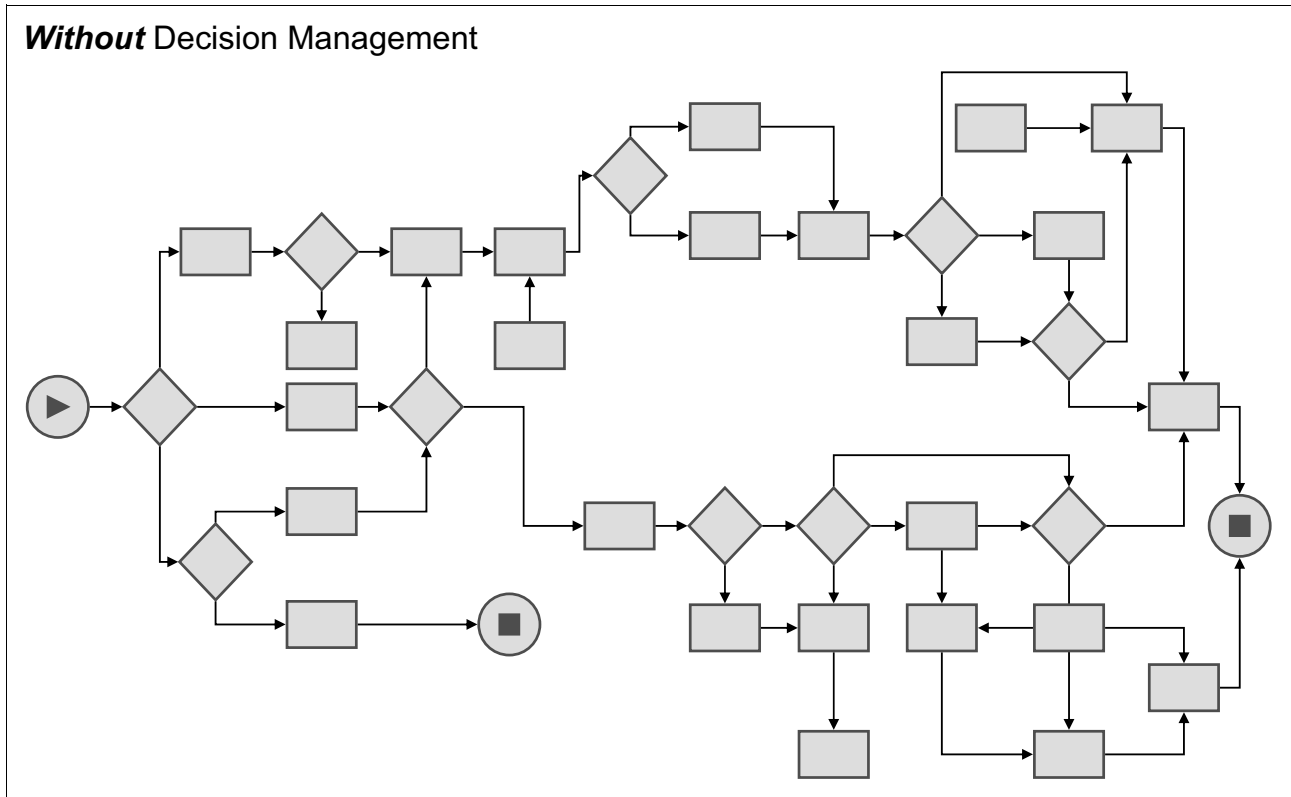


Figure 4-2 Complex business process with embedded decision logic

Adding decision management to the process can help remove the complexity of the process by simplifying parts of it that are decision services. At the same time, removing the decision logic from the process enables that logic to be reused elsewhere within the enterprise. Figure 4-3 on page 80 shows how it is possible to extract decision services portions of a business process to create a decision management solution. As you can see, this new solution would call on three decision services:

- ▶ Eligibility
- ▶ Risk Scoring
- ▶ Offers & Promotions

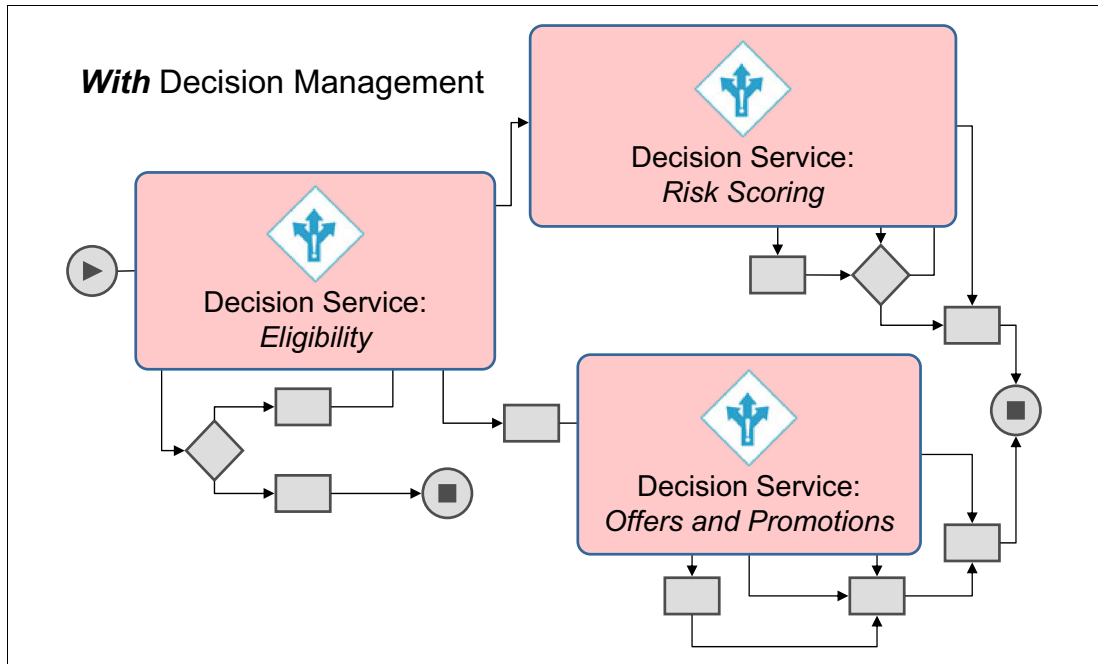


Figure 4-3 Business process simplified by extracting decision logic

4.2.1 Decision management usage patterns

Decision management can be used in the context of other applications or business processes in various ways. The examples in this book show a number of the most common usage patterns for how decision management can be used within the context of business process management.

Business process calling business rules

Business rules are typically called from a process in the same way any external, stateless service would be called: the process sends data to the decision service and receives a synchronous response. Because the decision service is stateless, all the information required by the decision service must be passed from the process. A number of design decisions must be made when using this pattern, such as what degree of processing should be done within the business process and what should be done within the rules. As a general principle, minimizing the number of calls between the business process and the decision service is desirable, because it minimizes performance hits. The design should also focus on keeping activities that are primarily process focussed, such as human workflow and system interactions, within the business process and those that are decision focussed within the decision service. Figure 4-4 on page 81 shows an example of how a business process can interact with the Decision Server at runtime.

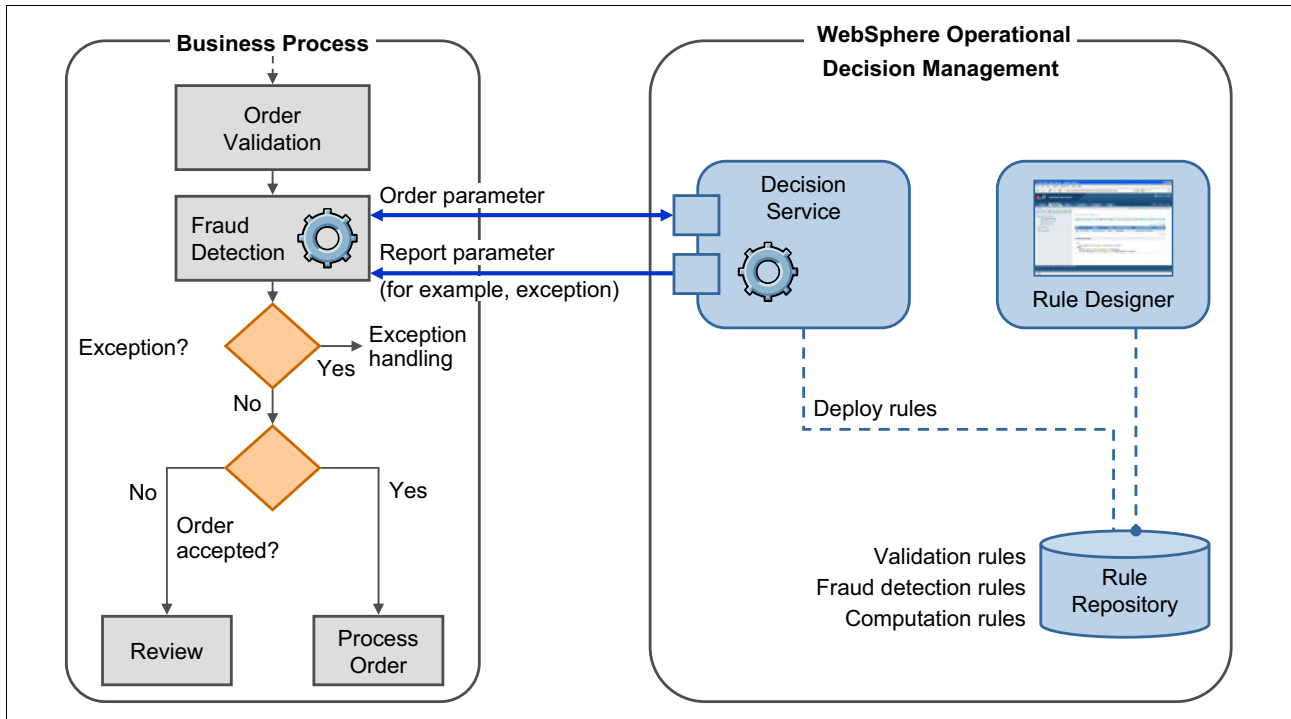


Figure 4-4 How rules and business processes interact

Triggering a process based on a business situation

Business events can be used to detect various situations (that is, conditions, such as possible fraud or a new opportunity) that require prompt action. It is common for any follow-up action resulting from such a business event to be a process. While other patterns exist, this sense-and-respond pattern is quite common, which is why we selected it for our warranty application. This pattern is shown in Figure 4-5.

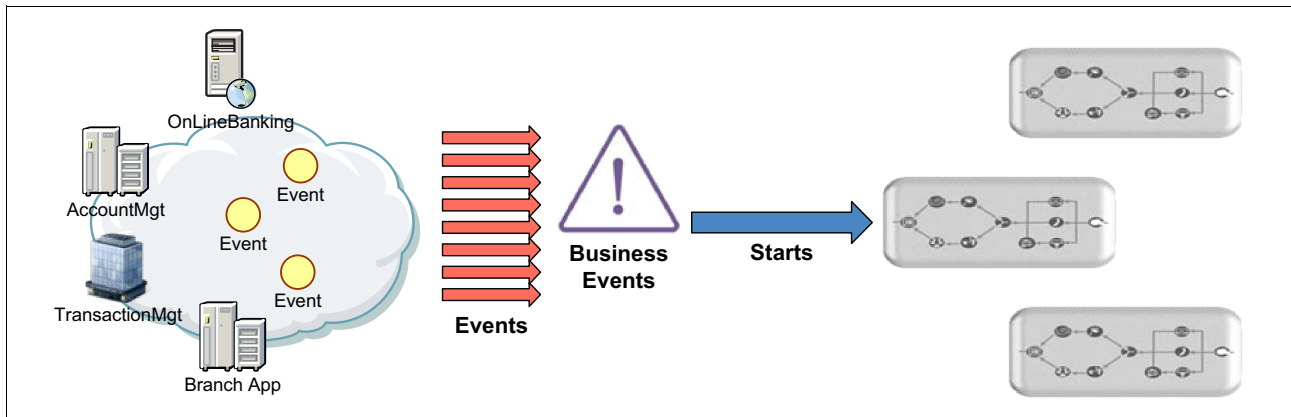


Figure 4-5 Common sense-and-respond pattern

4.3 WebSphere Operational Decision Management V8

IBM WebSphere Operational Decision Management V8.0, the next generation of business rules technology, delivers comprehensive automation and governance of operational decisions that control the actions of critical business systems.

It enables organizations to:

- ▶ Automate, govern, and improve operational decision making across business processes and applications for better business outcomes.
- ▶ Make more profitable decisions with real-time detection of opportunities and risks.
- ▶ Implement changes easily, reliably and securely in order to meet new market demands or policy requirements.

This section provides an overview of the components in the product and gives an overview of how each works.

Figure 4-6 shows the components that make up WebSphere Operational Decision Management V8.0.

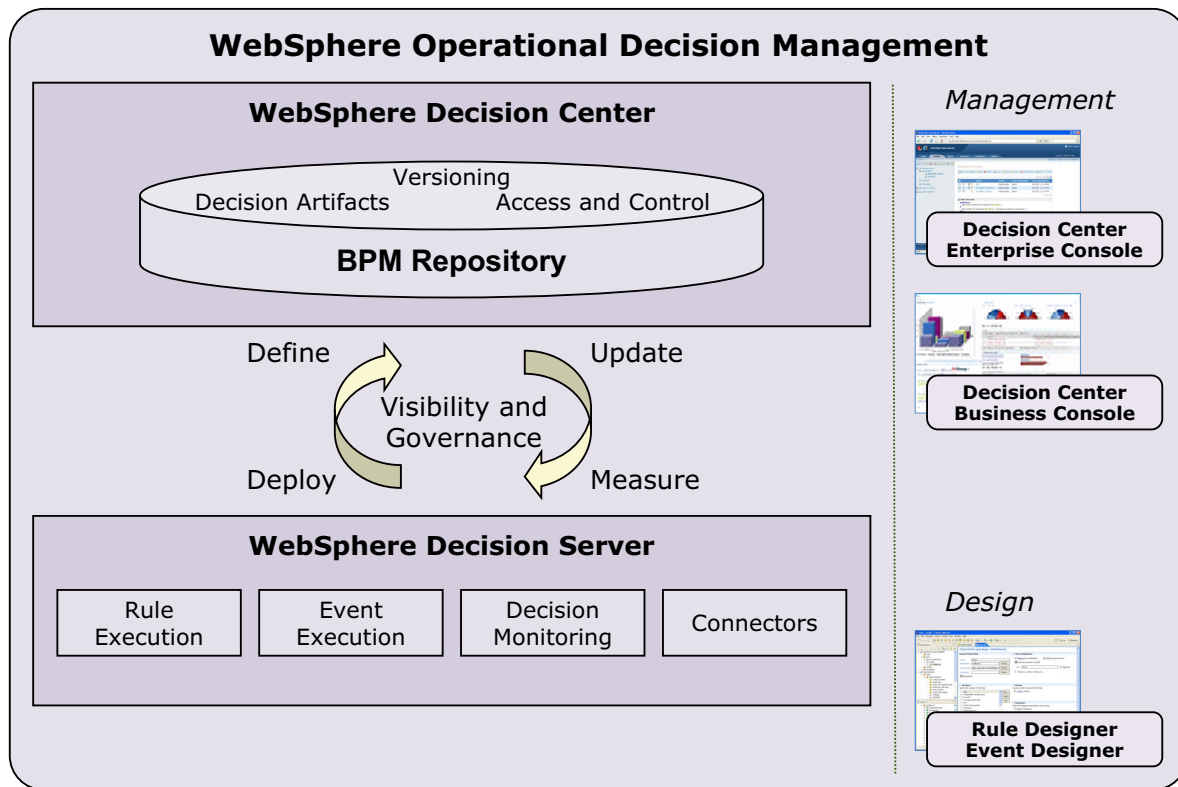


Figure 4-6 WebSphere Operational Decision Management v8

These components are described in more detail in the following sections.

4.3.1 Decision Server

Decision Server is the runtime component where business rules and event applications are installed and executed. Typically, Decision Server is hosted in a Java-based application server, and is packaged with WebSphere Application Server. You may also install and run it on other application servers or without an application server, albeit with limited capabilities.

4.3.2 Decision Center

Decision Center is a repository that contains the source of your rules. As such, it provides the governance capabilities and interfaces that enable business users to work directly with rules.

You may use WebSphere Operational Decision Management without using Decision Center. However, doing so means your business users will not be able to manage and change rules directly in the tool.

Both the Decision Center Business Console and the Enterprise Console enable business users to manage rules directly in WebSphere Operational Decision Management. Each console is summarized in the next two sections.

4.3.3 Decision Center Business Console

This console provides a socially-aware, web-based interface that enables business users to author, edit, organize, and search for rules in a collaborative environment. It is meant to coexist with the Enterprise Console, and offers the following features:

- ▶ Simplified navigation, to quickly access the projects and elements you want to work on.
- ▶ Simplified editing of action rules and decision tables.
- ▶ See what changes have been made to projects that interest you.
- ▶ See and post comments on recent changes made by other users, and attach files to these posts.
- ▶ Simplified search, available from a search box.
- ▶ Take snapshots of the current state of a project or its state at the time of a previous change.

Note: The term snapshot is used in the Business console and corresponds to the term baselines that is used in the Enterprise console.

4.3.4 Decision Center Enterprise Console

This console is more sophisticated than the business one, because it provides support for users who are involved in day-to-day management and testing of rules, such as:

- ▶ Managing branches
- ▶ Project settings
- ▶ Deployment, security, and repository administration
- ▶ Creating queries
- ▶ Project reports and rule analysis
- ▶ Testing and simulation
- ▶ Creating templates, variable sets, functions, technical rules, and resources

4.3.5 Rule Designer and Event Designer

Rule Designer and Event Designer are the development environments for developing rule and event applications. The following sections summarize each tool and describe the main activities that are required to make and deploy a project within each tool.

Rule Designer

Rule Designer is an Eclipse-based integrated development environment (IDE) for developing business rule applications. It is used by a Rules Developer who has Java development experience along with business rule development experience.

The highest level organizational unit within Rule Designer is a Rule Project, which consists of:

- ▶ An eXecutable Object Model (XOM) that represents the basic data structures upon which rules will be created.
- ▶ A Business Object Model (BOM) that maps the business vocabulary to the XOM.
- ▶ Rules that may exist in one of the following forms:
 - Business Action Language (BAL) rules, which are formatted as IF - THEN statements and are constructed using the vocabulary described in the BOM.
 - Decision Tables, which consist of one or more input columns and one or more action columns; typically, they are used instead of BAL rules when there are large numbers of rules that are similar in structure and only vary based on the parameters.
 - Decision Tree, which are rules that have built-in decision branches to deal with more complex decisions than can be easily represented by a decision table.
- ▶ Ruleflows confirm the order in which rules must execute, and they may contain branching logic. There must be at least one ruleflow in a project (for example the main ruleflow) because this is the entry point to rule execution.

Once you have developed these items, you need to deploy your rule project. First, you must create a RuleApp that references the Rule Projects you want to deploy. You then deploy this RuleApp to the Decision Server runtime.

Event Designer

Event Designer is an Eclipse-based integrated development environment (IDE) for developing situational awareness applications. It is used by an Event Developer who has Java development experience along with complex event processing experience.

The highest level organizational unit within the Event Designer is an Event Project. It consists of:

- ▶ Event - A message that is to be processed and can come from many sources, such as a database, a file system, a JMS queue, or a web service.
- ▶ Event Objects - Describe the data that is associated with the event. The event object is used to populate data inside a business object.
- ▶ Business Objects - Are temporary objects that are created at the time of an event in order to evaluate whether that event matches any event rules. Business objects can be made persistent, thus enabling rules to correlate events that happen over periods of time.
- ▶ Event Rule - Are IF - THEN statements written in Business Action Language (BAL) based on the vocabulary described in the Business Objects. They enable patterns of events to be matched.
- ▶ Action - An outgoing message that can be triggered upon the success of an event rule. It can send a message to many different channels, such as a database, a file, a JMS queue, or a web service.
- ▶ Action Object - Describes the data inside the business objects that are associated with an event.

4.4 Overview of business rules

Business rules can be represented in many forms, such as in a programming language or in company policy documents (written form). When derived from written forms, business rules can be centralized and automated, which allows them to be read and understood by someone

without having computer programming knowledge or experience. WebSphere Operational Decision Management combines these approaches to allow business users to create rules in structured English—or any other human-readable language that can be represented in the UTF character set, for that matter—and then deploy them so that they can be executed automatically. The following sections explain how to author and deploy business rules.

4.4.1 Building a Business Rule Project

Rule authoring is performed in Rule Designer, an Eclipse-based Integrated Development Environment (IDE). It consists of a series of sequential steps that are described here.

Create a Java Project with an eXecutable Object Model (XOM)

The eXecutable Object Model (XOM) is the base representation of the data structure that will govern the rules. For example, if a company is writing rules to process warranty claims, a data structure representing a warranty claim is needed. This data structure can be represented with one of the following:

- ▶ Java classes
- ▶ XML schema

For the warranty application, we will create the XOM using Java classes. Once the Rule Designer has been launched, you will want to switch to the Java perspective by selecting it in the top right corner of Rule Designer, as shown in Figure 4-7

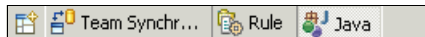


Figure 4-7 Switching to the Java perspective in Rule Designer

Next, you need to create a Java Project to hold your Java classes:

1. Select **File** → **New** → **Java Project** from the toolbar.
2. Name your project `WarrantyClaimsXOM`, and then click **Finish**.
3. To create a Java class, select **File** → **New** → **Class**.
4. Complete the dialog box for the class description as shown in Figure 4-8 on page 86, and click **Finish** when done.

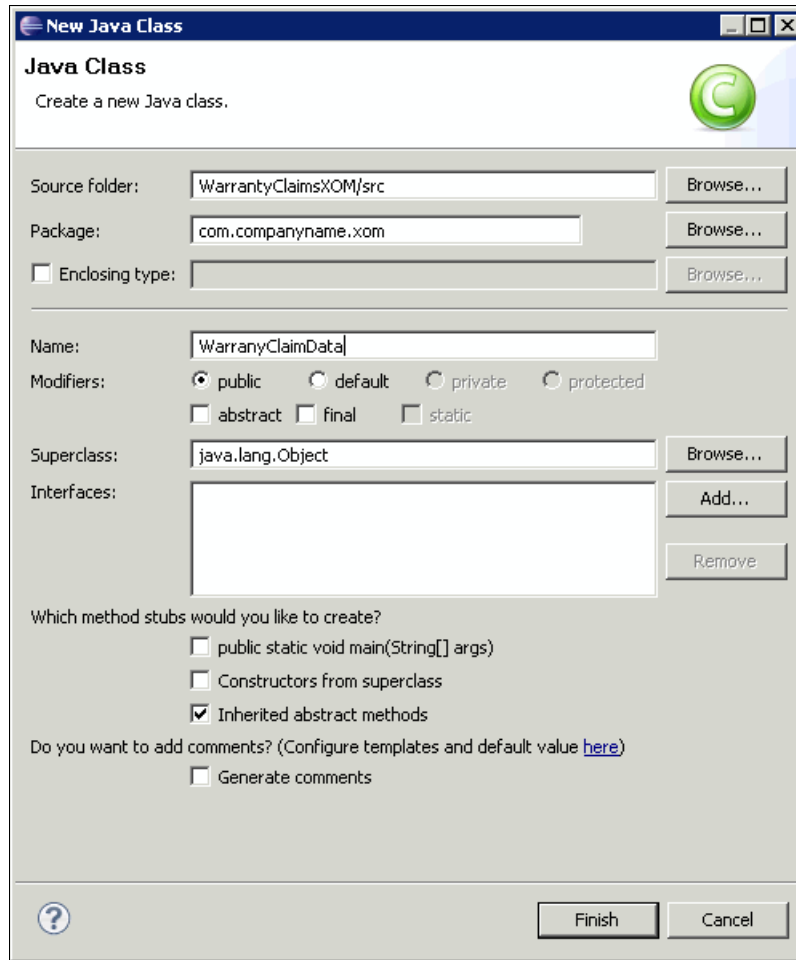


Figure 4-8 Creating a Java Class for the executable object model

5. Create the private fields for the class as shown in the top part of Figure 4-9 on page 87.

```

package com.companyname.xom;

public class WarrantyClaimData {
    private String serialNumber;
    private int secondsBetweenClaims;
    private String productType;
    private String problemType;
    private String advice;
    private double payoutAmount;
    private String action;
    private String status;

    public String getSerialNumber() {
        return serialNumber;
    }

    public void setSerialNumber(String serialNumber) {
        this.serialNumber = serialNumber;
    }

    public int getSecondsBetweenClaims() {
        return secondsBetweenClaims;
    }

    public void setSecondsBetweenClaims(int secondsBetweenClaims) {
        this.secondsBetweenClaims = secondsBetweenClaims;
    }

    public String getProductType() {
        return productType;
    }

    public void setProductType(String productType) {

```

Figure 4-9 Editing the Java class for the executable object model

6. The get and set methods for each field can be typed in directly, or you can generate them using the **Source** → **Generate Getters and Setters** option.

Your Java project now contains the XOM.

Create a Rule Project with a Business Object Model

Now you will create a Business Object Model (BOM). The BOM is the human-readable form of the XOM that is used to write rules. You will construct your BOM automatically from the XOM when you create your Rule Project.

You must switch to the Rule perspective as shown in Figure 4-10

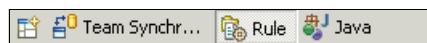


Figure 4-10 Switching to the Rule perspective in Rule Designer

1. Select **File** → **New** → **Rule Project** from the toolbar.
2. Select **Rule Project with a BOM** option and click **Next**.
3. Name the project `WarrantyClaimsRules` and click **Next**.
4. In the New Rule Project dialog, select the **WarrantyClaimsXOM** project created in the previous section, and click **Finish** when done, as shown in Figure 4-11 on page 88.

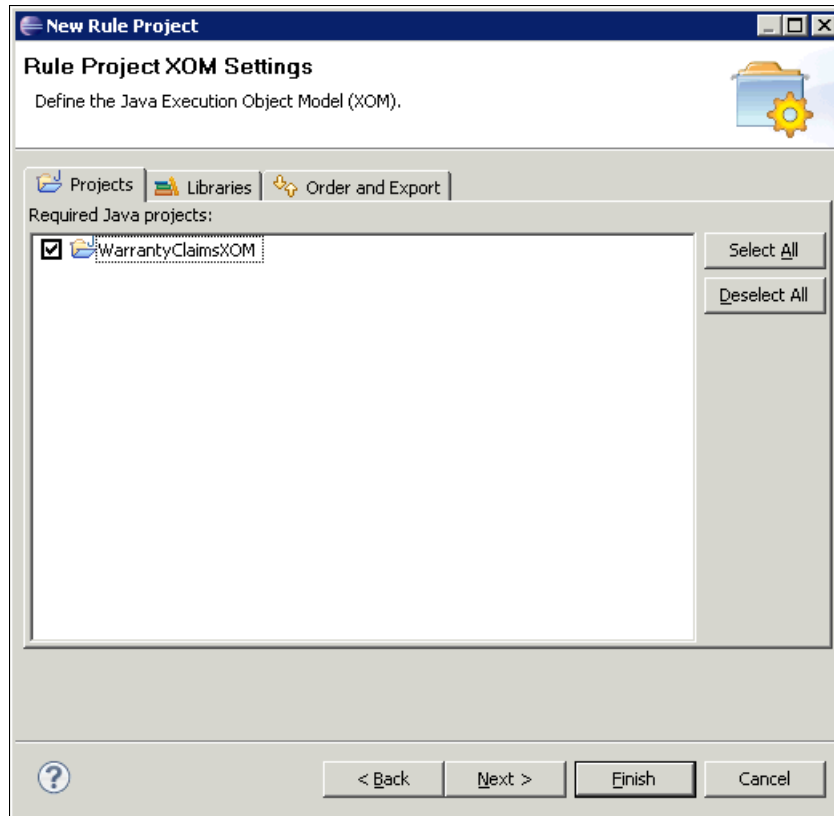


Figure 4-11 Attaching the executable object model to a Rule Project

You created your Rule Project containing the BOM.

To understand where you are in the process of creating a functioning Rule Project, you may switch to the tabbed pane at the bottom of Rule Designer labelled Rule Project Map. This shows a road map indicating which elements in your Rule Project have been created and which remain to be done. Your diagram should look like the one shown in Figure 4-12.

Tip: The project must be selected in the Rule Explorer pane in order for the Rule Project Map to be visible.

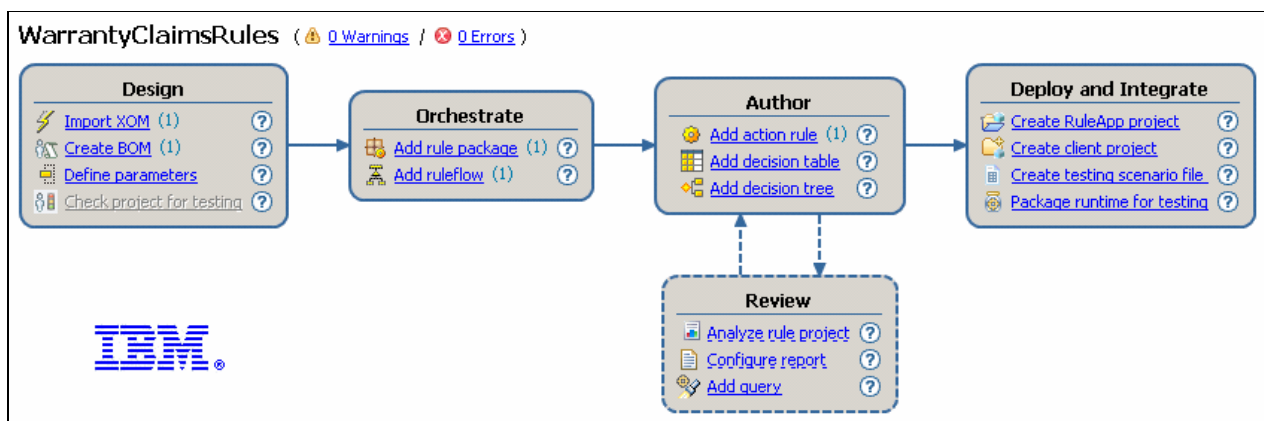


Figure 4-12 Overview of the steps involved in creation of a rule project


```

if
    the product type of 'warranty claim' starts with "A"
then
    set the status of 'warranty claim' to "automatic";
    print "A";
else
    set the status of 'warranty claim' to "manual";
    print "M";

```

Figure 4-14 The warranty claim automation rule

4. Press Ctrl+S to save your rule.

This rule will set the status of the warranty claim to automatic if the product type starts with the letter A; otherwise the status will be manual. The print function was enabled for debugging purposes to ensure our rule executes correctly.

4.4.2 Rule execution

Before a rule can be executed, Decision Server must be running and the rules must be deployed. It is not possible to deploy a Rule Project directly; instead, a RuleApp project must be created that references one or more Rule Projects, and you deploy it. Once the RuleApp is deployed, test it to ensure that it functions as intended.

Rule deployment

To deploy a Rule Project:

1. Start the Decision Server from the Windows Start menu by choosing **Start** → **All Programs** → **IBM** → **WebSphere Operational Decision Management V8.0** → **Sample Server** → **Start Server**. (The first time this is done, it will take a few minutes for the server profile to be created.)
2. Select **File** → **New** → **Project** from the toolbar, and then select **RuleApp Project** to create a new RuleApp.
3. Name the project `WarrantyClaimsRuleApp` and click **Next**.
4. Next, click **Add**, select **WarrantyClaimsRules**, click **OK**, and then click **Next** when done.
5. Click **Finish** to create your RuleApp project, which references the Rule project and is now ready to be deployed on the server.
6. Ensure that the Decision Server has started by checking its status in the Command Prompt window that opened when the server was launched.
7. In the Rule Explorer view, right-click **WarrantyClaimsRuleApp** and choose **RuleApp** → **Deploy** as shown in Figure 4-15 on page 91.

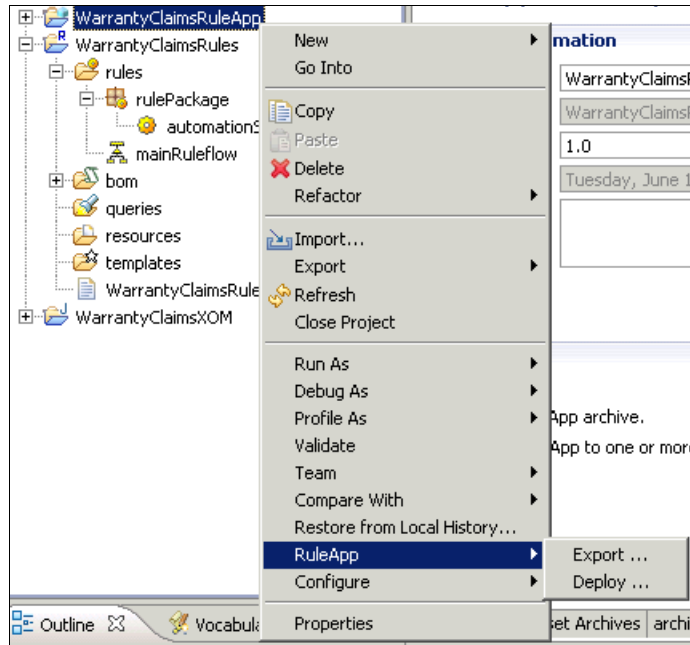


Figure 4-15 Creating a new RuleApp in Rule Designer

8. Click **Next** when asked about the RuleApp version, as this step is not important at this stage.
9. Enter the details for your Decision Server; typically the default values are as follows:
 - URL: `http://localhost:9080/res`
 - Login: `resAdmin`
 - Password: `resAdmin`
10. Click **Finish** to initiate the deployment.
11. If the RuleApp deploys successfully, the Console tab in Rule Designer should look as in Figure 4-16.

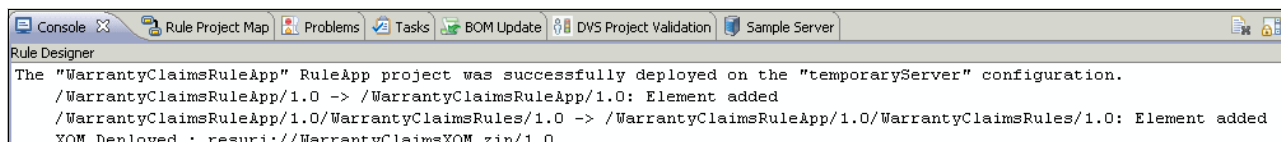


Figure 4-16 Rule designer console showing successful deployment of a RuleApp

Rule testing

Now test the rule that you just deployed:

1. Select **Run** → **Run Configurations** from the Rule Designer toolbar.
2. Click **WarrantyClaimsRules** under Rule Project, and select the **Parameters & Arguments** tab.
3. Click **ruleData** and press the Edit Value button.
4. Edit the parameter value so that it appears as shown in Figure 4-17 on page 92.

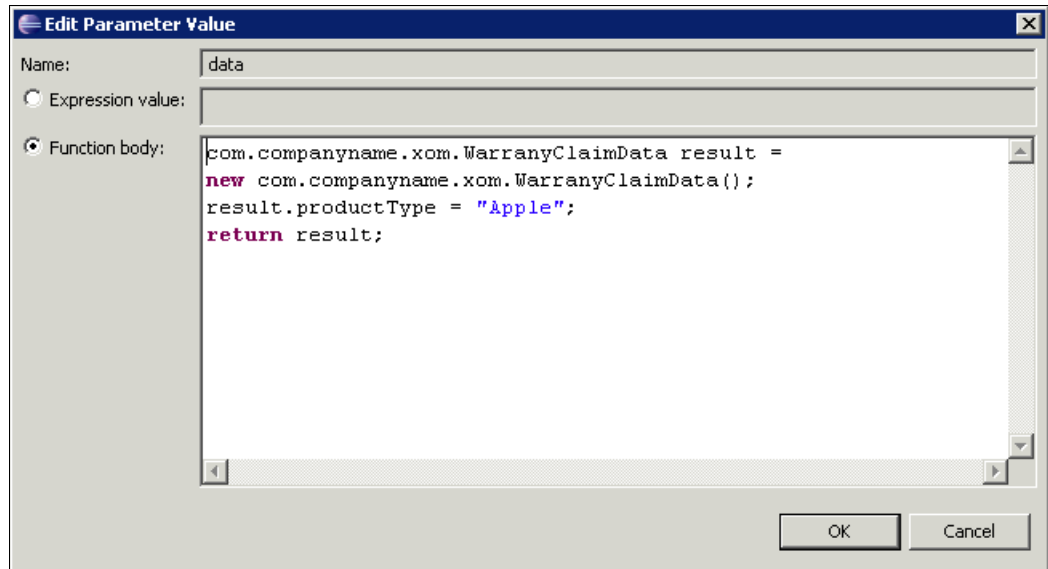


Figure 4-17 Creating a test object in Rule Designer

5. Click **OK** when done, and then click **Run** to test your rule.

The rule will execute, and if it completes successfully, your Console tab in Rule Designer should display the letter A. This is the result of the debug statement that indicates the product type should have a status of automatic.

4.5 Overview of Event Rules

Event Rules are similar in many ways to the business rules that are described in 4.4, “Overview of business rules” on page 84. The key difference between them is that event rules always have a time component, which could be phrases such as “occurred within the last week” or “did not occur within 2 hours of. . .”

Table 4-1 shows the various artifacts that will make up our warranty project.

Table 4-1 Artifacts of the Event Rule Project

Name	Type	Description
EventReceived	Event	The incoming event received from a web service, which indicates a single warranty claim has been made
EventReceivedObject	Event Object	A description of the data in the EventReceived event
WarrantyBusinessObject	Business Object	An object that the event object data copies and enriches when the event rule is evaluated
WarrantyHistoryBusinessObject	Business Object	An object that accumulates historical events for users to correlate over time

Name	Type	Description
StartWarrantyEventInBPM	Action	The action generated by importing the WSDL of the BPM process, which starts the warranty process
StartWarrantyEventInBPMRequest	Action Object	A description of the data associated with the StartWarrantyEventInBPM action that will be sent to the process
MultipleWarrantyEvents	Event Rule	An event rule that will detect a number of claims for the same serial number, if the warranty claims are processed in quick succession

4.5.1 Building an Event Project

Event Designer is a Decision Server Events component that supports the definition of the metadata layer required for business event processing (BEP). You can use Event Designer to create all the building blocks for your application, including events, business objects, actions, and event rules.

This section highlights the sequential steps for you to create executable rules using Event Designer.

Create an Event Project

The Event Project for the sample warranty application is tied to the business process described in Chapter 5, “Business process management with Business Process Manager” on page 113. This business process defines a web service interface that is used by our Event Project.

Note: It is important to save any item that is created whenever it is changed. The project will not synchronize until a save is done, and if you attempt one without first saving your work, many errors may result.

To create a new Event Project based upon an existing web service:

1. Select **File** → **New** → **Event Project** from the toolbar.
2. Name your project **WarrantyEvents**, and click **Finish** when done.

Note: You will receive an error stating that “At least one field must have a definition.” You will learn how to resolve this error in a later section.

Now import the Web Service Definition Language (WSDL) file for the web service that will enable you to invoke the business process when multiple warranty claims are detected.

1. Right-click the newly created **WarrantyEvents** and select **Import**.
2. In the import dialog, expand **Event Designer** and select **WSDL into Event project**.
3. Click **Next** and enter the following address to specify where to import the WSDL from:

<http://ibmbpm:9080/teamworks/webservices/RWP/StartWarrantyEventsWS.tws?WSDL>

Note: This address is also used in Chapter 5, “Business process management with Business Process Manager” on page 113.

Your dialog should be similar to what is shown in Figure 4-18.

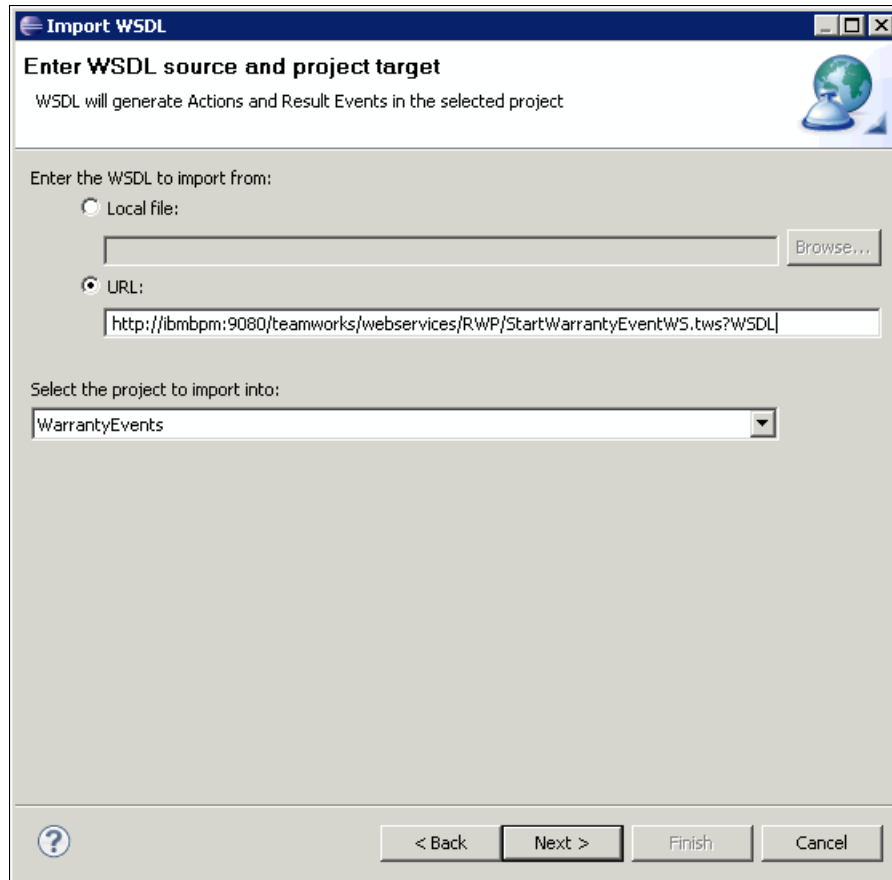


Figure 4-18 Importing a WSDL into Event Designer

4. Click **Next** followed by **Finish**.

The event project has now been created with some minimal artifacts that will trigger the BPM process when the events match the defined pattern.

Create an event

Now describe the creation of the incoming event. You need to define the event and the event object that describes the event’s data.

1. Right-click **WarrantyEvents** and select **New** → **Event** from the pop-up dialog.
2. Leave the next panel as **Create a blank event** and click **Next**.
3. Name the event `EventReceived` and click **Next**.
4. Select **SOAP** as the Event Connection type and click **Finish**.
5. Click **Save** to create your event.

It does not have a data model associated with it, so you need to define the Event Object that will hold the event's data:

1. Select **EventReceived** (your new event), and on the right side of the Event editor in the Event Objects section, click **Add**.
2. Specify **Add a new blank event object** and click **Next**.
3. Name the event object `EventReceivedObject` and click **Finish**.
4. In the `EventReceivedObject` editor, go to the Fields section and click **Add**.
5. Specify a Field name of `SerialNumber` with type `String`.
6. Click **Finish**.
7. In the Fields section, click **Add**.
8. Specify a Field name of `eventTime` with type `DateTime`.
9. Click **Finish**.
10. Click **Save All**.

The resulting Event Object fields should look as shown in Figure 4-19.

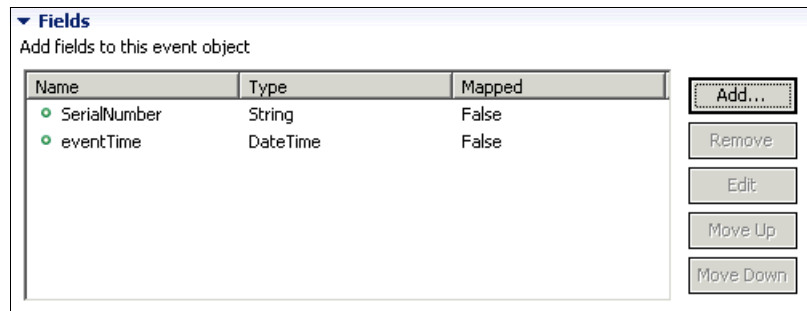


Figure 4-19 Event Object fields after they have been defined in Event Designer

Create a business object

Business objects are representations of objects such as Customer, Product, or Order. The primary purpose of a business object is to hold the data that is evaluated by event rules at run time. The data that populates a business object does exist in the enterprise applications, but it typically does not exist in a single record or structure. The fields in a business object can be populated from various sources.

A business object as defined to Decision Server Events is the ideal representation of the business object, because it serves as a template for sharing data, containing a number of fields that can be populated from many different business applications across the enterprise. Each field includes the name and type of information, as well as an optional definition for how to get a value, if none is supplied with the event.

You will create two business objects in your warranty project: one to hold the event data of the current event and another to accumulate historical data so that it is possible to compare events over time.

1. Right-click **WarrantyEvents** and select **New** → **Business Object**.
2. Enter the name `WarrantyHistoryBusinessObject` and click **Next**.
3. Select the option **Create an accumulating array that retains the values of fields from previous events** and click **Next**.
4. Enter a maximum number of instances of 2 and click **Next**.

5. Select **EventReceivedObject** and click **Finish**.

It is now necessary to modify the created business object, because we do not wish to accumulate data on all of the fields of the event object.

6. In the Fields section of the newly created business object, select **SerialNumber** and click **Remove**.
7. Click **Save**.

Now create another business object that holds the data that will be passed to the outgoing web service that launches the process in response to the situation being detected.

1. Right-click **WarrantyEvents** and issue **New** → **Business Object**.
2. Enter the name `WarrantyBusinessObject` and click **Next**.
3. Leave the option **Start with a blank business object** selected and click **Finish**.
4. In the Overview tab of the newly created business object, in the Fields section click **Add**.
5. Specify a Field name of `SerialNumber` and a Data type of `String` and click **Finish**.
6. In the Fields section of the newly created business object click **Add**.
7. Specify a Field name of `TimeDifference` and a Data type of `Integer` and click **Finish**.
8. In the Fields section of the newly created business object click **Add**.
9. Specify a Field name of `LastEventTime` and a Data type of `DateTime` and click **Finish**.
10. Double-click the **TimeDifference** field to edit it, choose a definition type of **Javascript** and fill in the expression box as follows:

Example 4-1 TimeDifference function

```
function timediff(f) {  
    var diff = f[1] - f[0];  
    return Math.floor(diff/1000);  
}
```

```
timediff(WarrantyHistoryBusinessObject.eventTime);
```

11. The Field definition should look like Figure 4-20 on page 97.
12. Click **Save**.

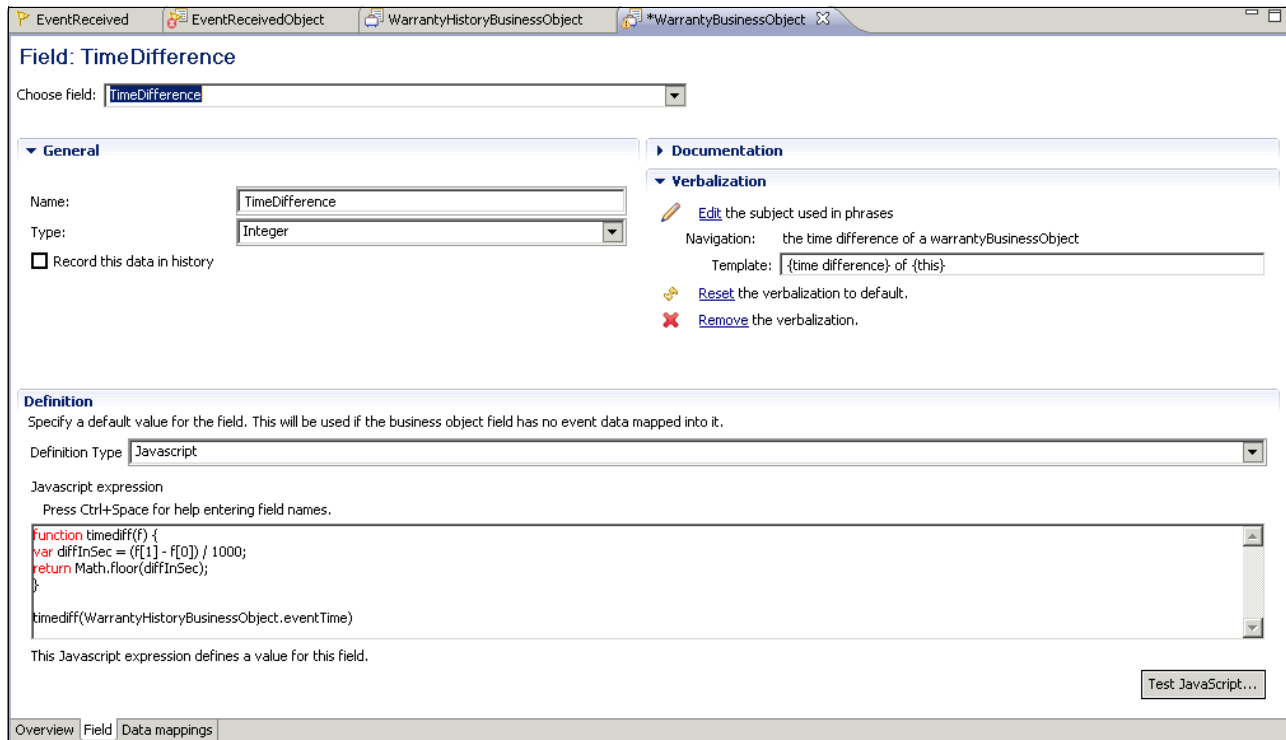


Figure 4-20 Javascript for the TimeDifference field

13. Click the **Overview** tab to return to the list of fields.
14. Double-click the **LastEventTime** field to edit it, choose a definition type of Javascript and fill in the expression box as follows:

Example 4-2 LastEventTime function

```
function lastTime(f) {
    return f[1];
}

lastTime(WarrantyHistoryBusinessObject.eventTime);
```

15. The field definition should now look like Figure 4-21 on page 98.
16. Click **Save**.

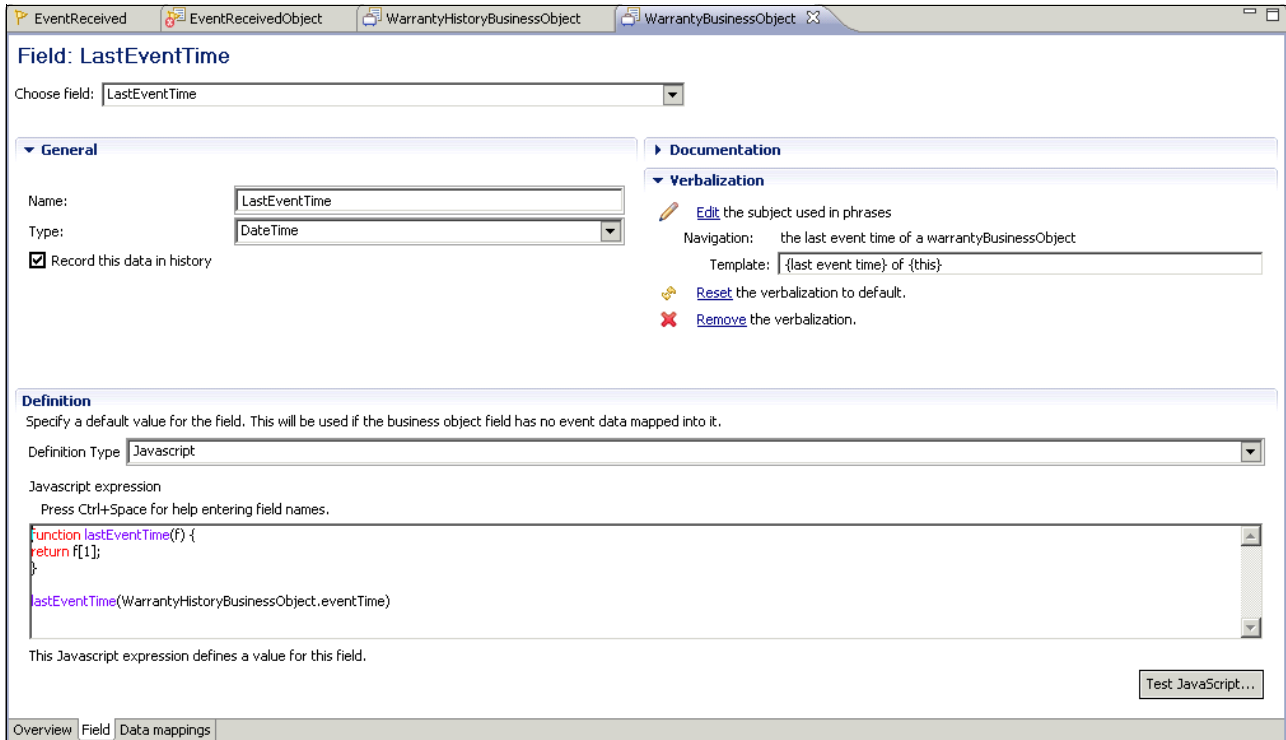


Figure 4-21 Javascript for the LastEventTime field

The creation of the Business Objects is now complete.

Map the Event Object to the Business Object

When your Event Object was created, it contained errors because there was no Business Object to populate. In the previous section, however, you created your Business Objects, so now you can map the fields in your Event Object to your Business Objects.

1. Double-click **EventReceivedObject**.
2. In the Fields section in the Overview tab, select **SerialNumber** and click **Remove**.
3. In the Field Constructors section in the Overview pane click **Add**.
4. Select the **WarrantyBusinessObject** and click **Finish**.
5. In the Field section, double-click the **SerialNumber**.
6. Select a Definition Type of **Field** and select the Event object field **SerialNumber**.
7. Click **Save**.

The mapping of the Event Object to the Business Objects is complete, and your mapping should look like Figure 4-22

Field Constructors				
Add business objects that will receive data from the fields in this event object:				
Business Object	Field	Data Type	Definition Type	Definition Value
WarrantyBusinessObject	SerialNumber	String	Field	SerialNumber
WarrantyBusinessObject	TimeDifference	Integer	None	
WarrantyBusinessObject	LastEventTime	DateTime	None	
WarrantyHistoryBusinessObject	eventTime	DateTime	Field	eventTime

Figure 4-22 Field constructors for the completed Business Object

Map the Business Object to the Action Object

Just as it was necessary to map the input from an Event Object to the Business Object, it is necessary to map the output from a Business Object to an Action Object.

1. Double-click the Action Object **StartWarrantyEventInBPMRequest**.
2. Double-click the field **impl:SerialNumber**.
3. In the Definition section select **Field** as the Definition Type.
4. Select the Business Object **WarrantyBusinessObject** and the Business Object field **SerialNumber**.
5. Select the **Overview** tab to return to the list of fields.
6. Double-click the field **impl:TimeBetweenEvents**.
7. In the Definition section select **Field** as the Definition Type.
8. Select the Business Object **WarrantyBusinessObject** and the Business Object field **TimeDifference**.
9. Select the **Overview** tab to return to the list of fields.
10. Double-click the field **impl:TimeOfLastEvent**.
11. In the Definition section select **Field** as the Definition Type.
12. Select the Business Object **WarrantyBusinessObject** and the Business Object field **LastEventTime**.
13. Click **Save**.

The mapping of the Business Object to the Event Object is complete, and your mapping should look like Figure 4-23.

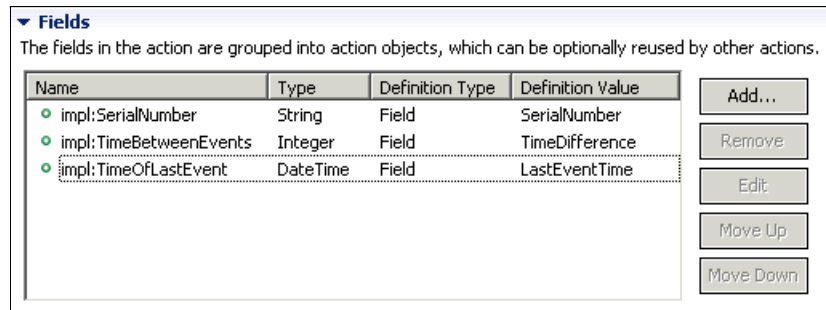


Figure 4-23 Completed Action Object mappings

Create an Event Rule

The only thing missing to make this a functioning event project indicating a response to the situation being detected is an event rule. Event rules are the conditions under which, based upon a pattern of events, an action should fire. To create an event rule:

1. Right-click **WarrantyEvents** and select **New** → **Event Rule**.
2. Name the rule **MultipleWarrantyEvents** and click **Next**.
3. Select the event **eventReceived** and click **Next**.
4. Specify the context relationship as **serial number of the WarrantyBusinessObject** and click **Finish**.
5. Create the rule by typing in the content that appears in Figure 4-24 on page 100.

Note: As you type the rule, you will be presented with suggestions based on the object model. You can activate this list of options any time within the rule by pressing Ctrl+Space.

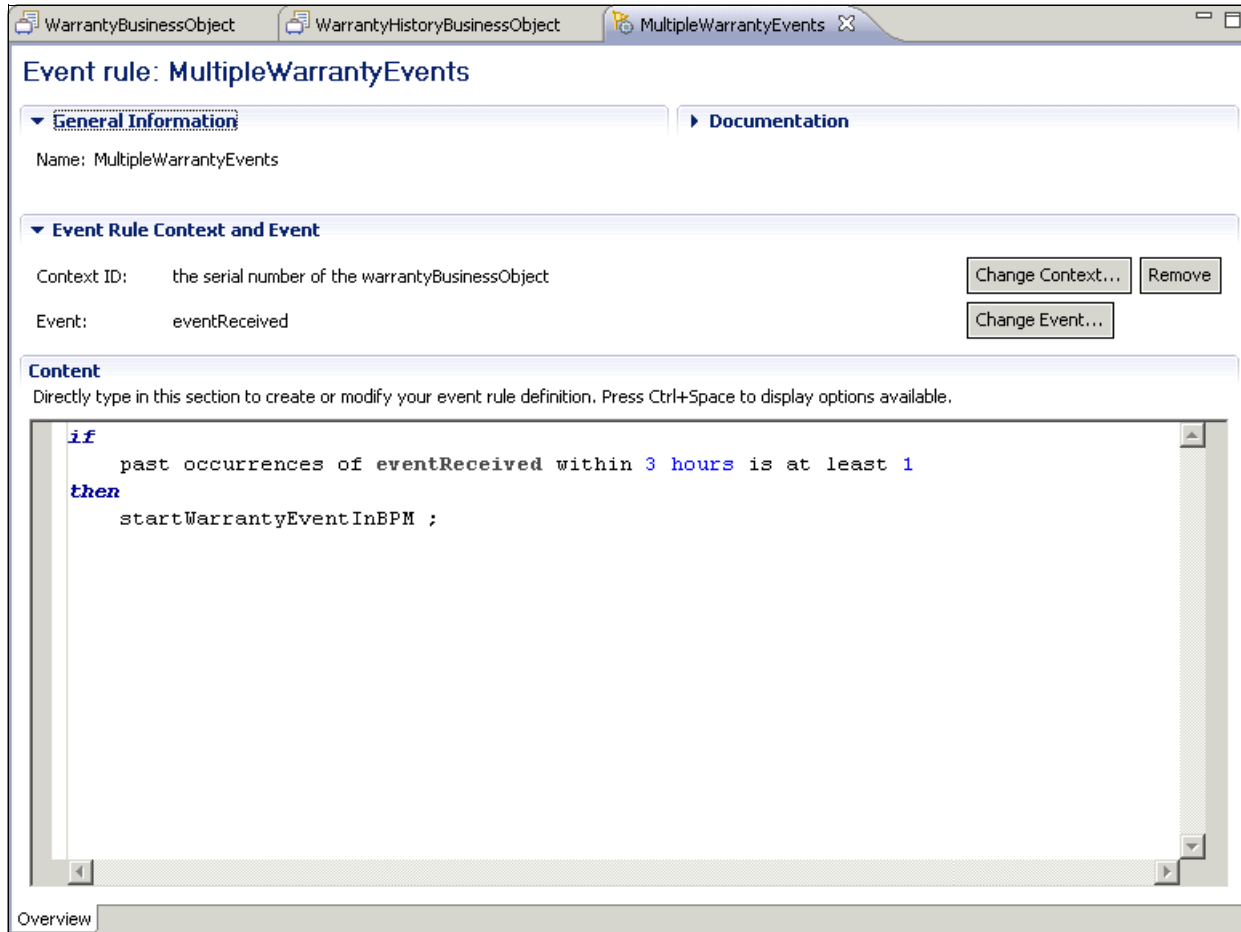


Figure 4-24 Event Rule to detect multiple claims within a short period of time

6. Click **Save**.

The Event Project is complete, and in order to test that it works, it must be deployed.

4.5.2 Event deployment

First, you deploy the rule as follows:

1. Start the Decision Server by selecting (from the Start Menu if using Windows) **Start** → **All Programs** → **IBM** → **WebSphere Operational Decision Management V8.0** → **Sample Server** → **Start Server**.

Note: The first time this is done, it will take a few minutes for the server profile to be created.)

2. Right-click **WarrantyEvents** and issue **Deploy**.

3. Select **Deploy all assets** and click **Next**.
4. Enter the credentials of your decision server configuration (they should appear similar to those in Figure 4-25).

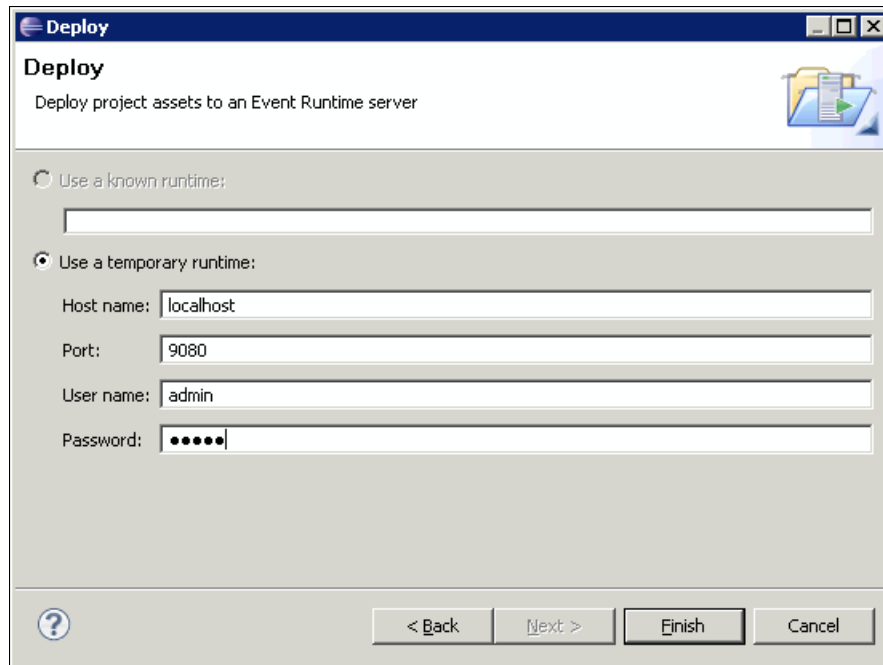


Figure 4-25 Event deployment server configuration dialog

5. Click **Finish** and wait for deployment to terminate.
6. After receiving the successful deployment message, click **OK**.
Now check the WSDL to ensure that the event project deployed correctly.
7. Open a web browser and enter the following address in the address bar:
<http://localhost:9080/wbecasoap/SOAPEventWSDL>

The browser should display a WSDL like the one in Figure 4-26 on page 102.

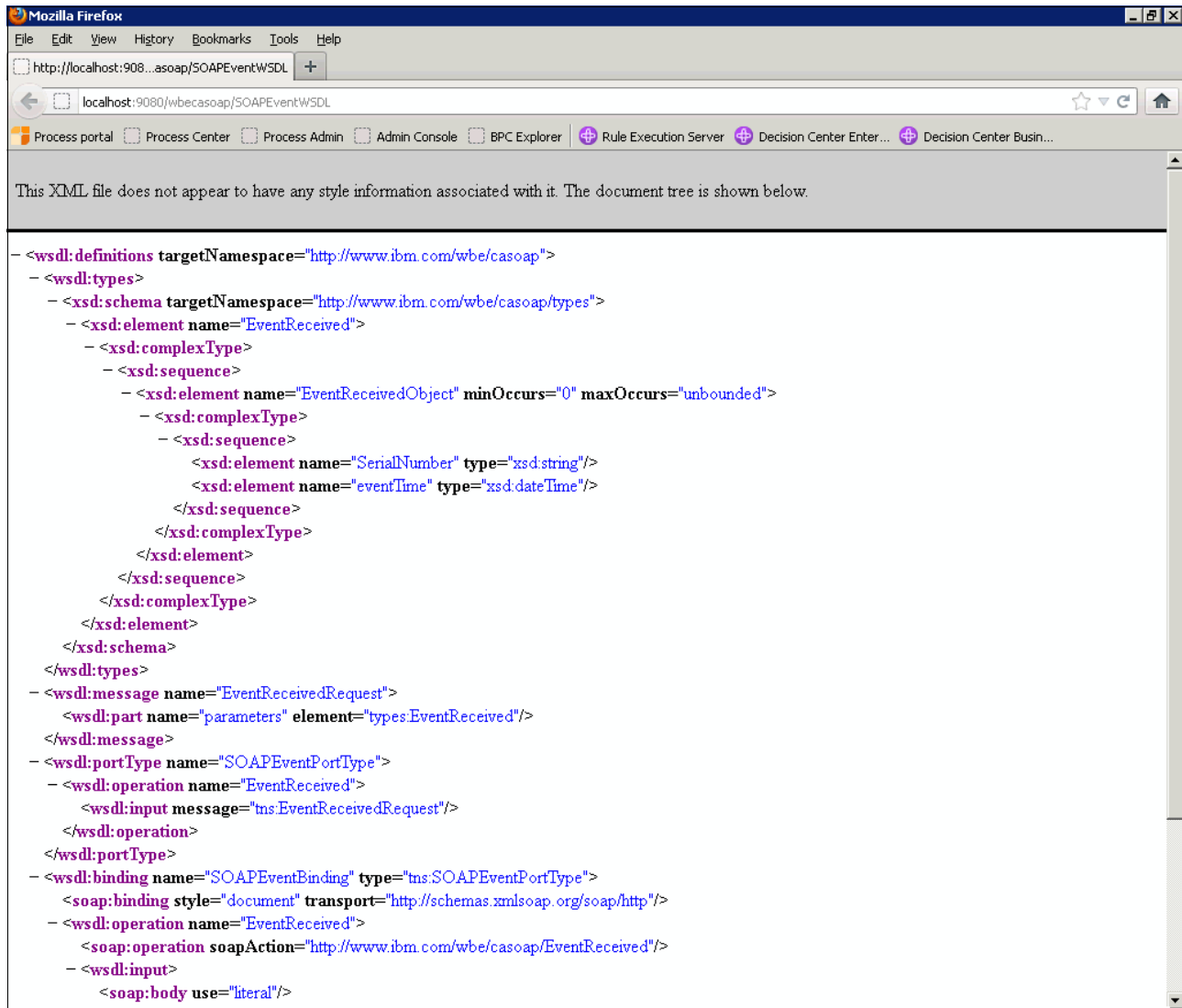


Figure 4-26 Event WSDL that is exposed so that the event runtime can receive external events

4.5.3 Event testing

WebSphere Operational Decision Management has a simple web-based testing interface to check that the event project is working. The testing interface is available using Business Space.

Note: In the sample environment used to support this project, we leveraged Business Space via IBM Business Process Manager, which was installed as part of Process Portal. If you do not have IBM Business Process Manager installed, you may skip the initial steps and log in directly to Business Space.

Set up Business Space

1. To access Business Space, or Process Portal if that is what you have installed, open a browser and enter in the address bar the address:

<http://localhost:9443/mum/enabler>

2. Enter the administration user name and password and click **Login**.
3. If your initial login sends you to the home page of Process Portal, in the top right corner click the round world symbol shown in Figure 4-27.

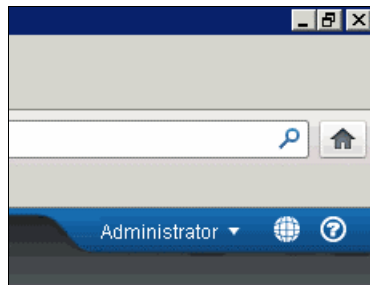


Figure 4-27 Changing from Process Portal to Business Space

4. Click the link **Welcome to Business Space**.
5. Select **Manage Spaces** → **Create Space**.
6. Name the new space **Event Testing** and click **Save**.
7. Create a page for the space, provide a name for it and leave the “Create an empty page” radio button selected. Click **OK**.
8. Click the **Event Testing** space you just created.
9. Click the **Edit page** link in the right corner.
10. Select **Events** in the drop-down list.
11. Drag the **Event Tester widget** onto the new blank page as shown in Figure 4-28.

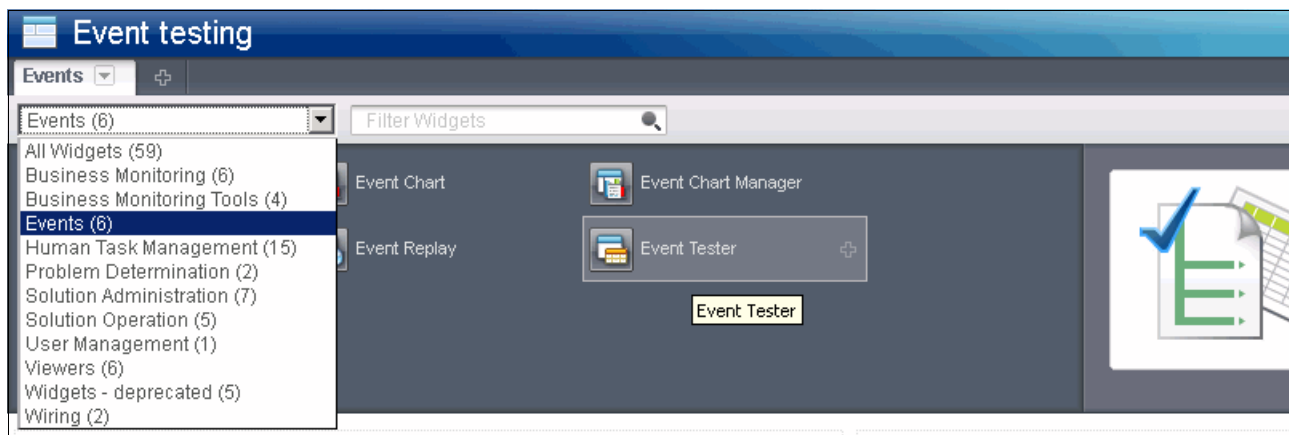


Figure 4-28 Event testing page

12. Click **Save** and then **Finish Editing**.

The Event testing widget enables you to send event data to the event runtime. Select the event template for EventReceived and try entering different events. As soon as you enter two events within three hours of each other for a warranty claim containing the same serial number, the action will fire. See Figure 4-29 on page 104.

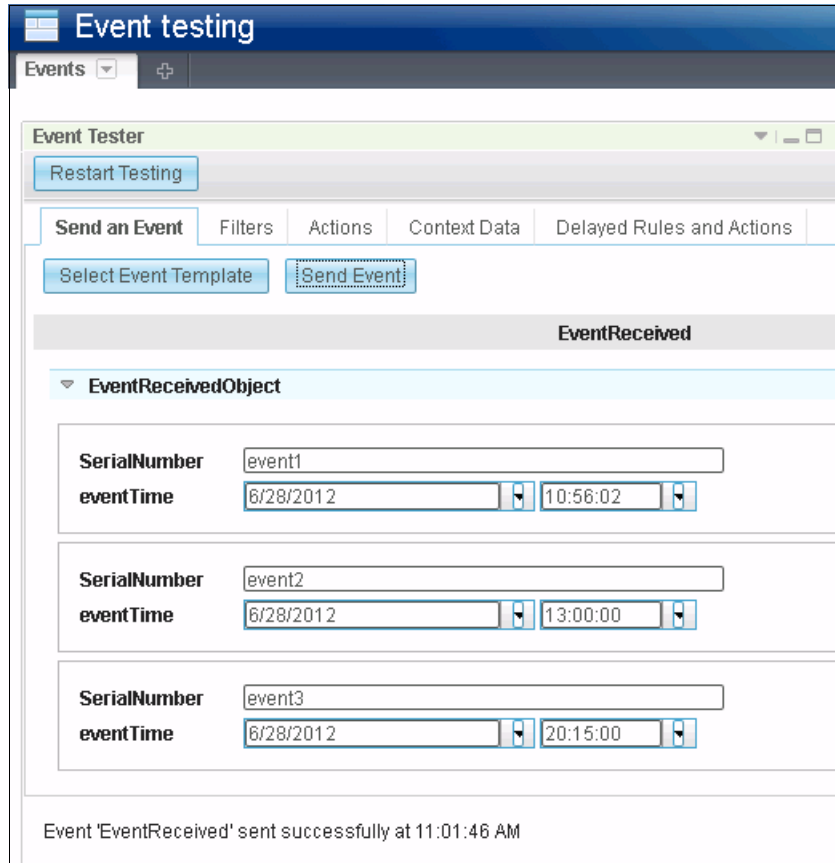


Figure 4-29 Event tester

This action will be visible in the Business Events testing interface. Additionally, as the event action has been wired to the business process, you can open Process Designer by selecting **Start** → **All Programs** → **IBM** → **Process Designer** → **Process Designer v8.0**, logging in as admin/admin and going to the inspector view to confirm the new processes that have been launched.

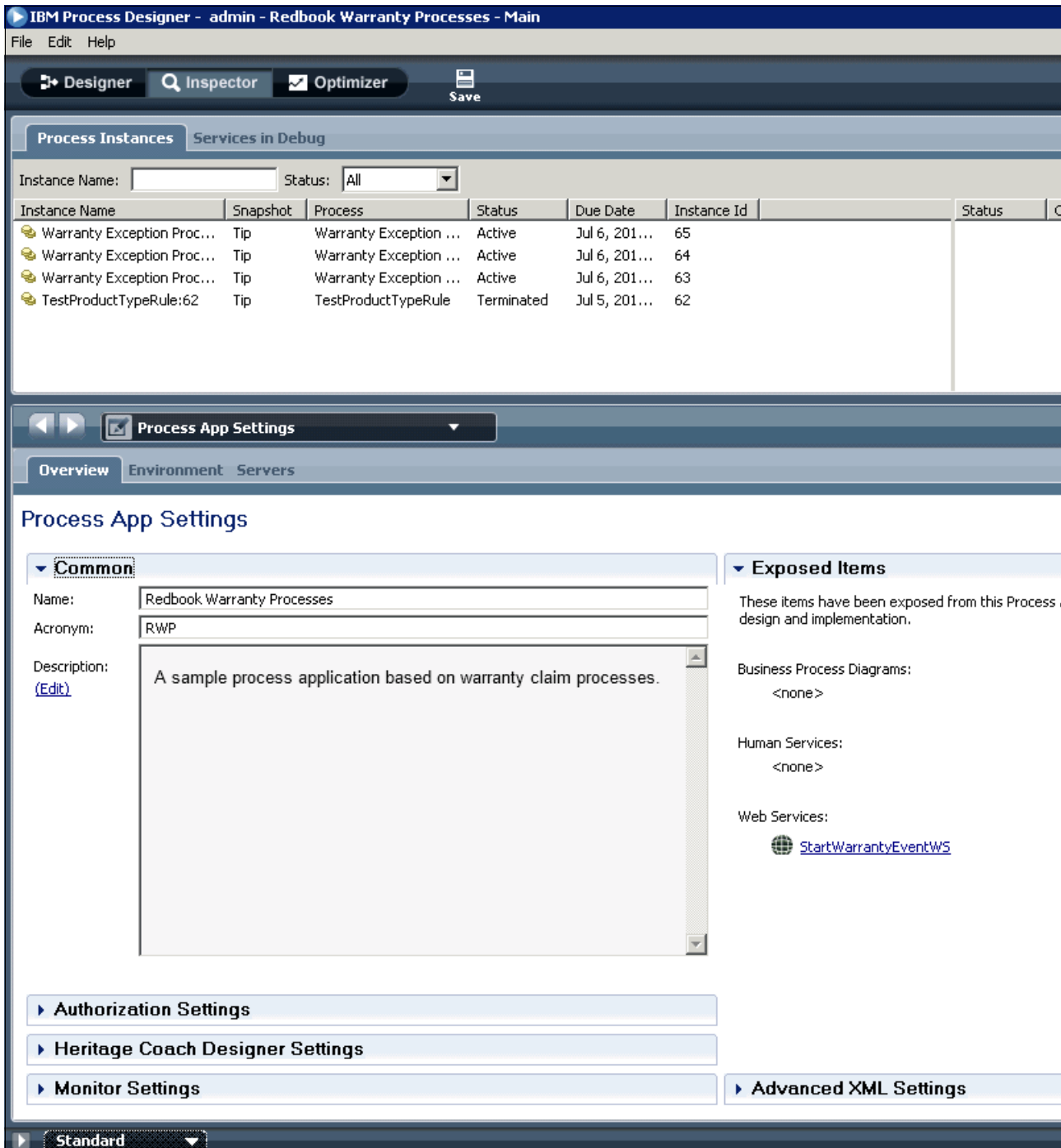


Figure 4-30 Viewing processes that have been triggered from events

4.6 Business management of rules

The core business benefit of business rules and event rules is that they can be managed by the business users. This requires a large cultural change for many organizations, and many in IT may resist giving power to make system changes directly to the business users. To address these concerns it is important to discuss how WebSphere Operational Decision Management provides the ability for the business users to manage rules in a controlled manner. In this

section we describe the principles of the rule governance model, which can be customized to provide the level of control desired by an organization and will also show the procedure for deploying a rule application so that it can be managed effectively.

4.6.1 Rule governance and promotion model

We define rule governance as the procedures and mechanisms that are put in place around rule maintenance. These procedures and mechanisms ensure that rules are performed in a controllable manner. The core component of WebSphere Operational Decision Management, which enforces rule governance, is the Decision Center via the rule repository. Every rule inside this repository has a status (such as New, Defined, Tested, Rejected), and each is defined by the organization building the rules. In addition, the organization will define roles that are authorized to interact with the rule management system given the level of access control granted to each user. Using a combination of roles and statuses, it is possible to construct a state diagram that describes at a high level the rule governance model. An example of such a model is shown in Figure 4-31.

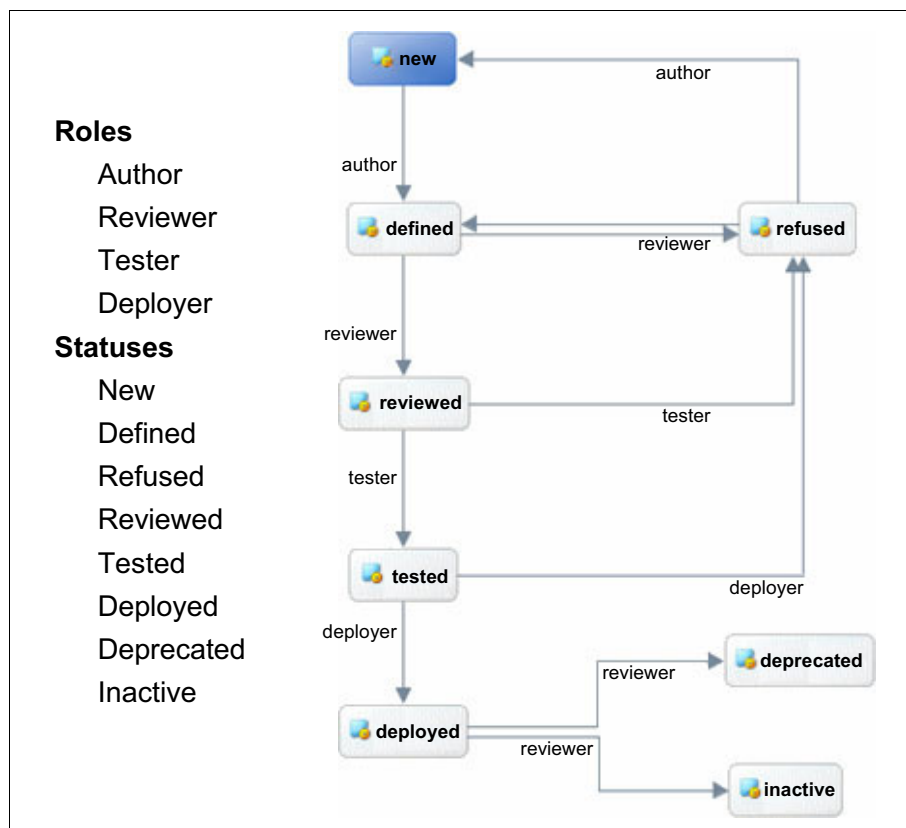


Figure 4-31 An example of a rule governance and promotion model

As seen in the diagram, each role has a defined set of transitions that it can perform from one status to another. In addition, it is possible to restrict each role in terms of capabilities. For example, certain authors may only have the ability to change certain groups of rules. It is also possible to define templates that restrict which parts of a rule can be modified.

4.6.2 Using Decision Center

Decision Center has a repository where rules are stored for business user interaction. It is the role of the business rule developer and event rule developer to synchronize in the Decision

Center the rules each developed in order to make them available to the business user community. After the rule or event projects have been deployed, a business user can log onto the Decision Center via a web interface to manage their rules.

There are two web interfaces:

- ▶ Decision Center Enterprise Console
- ▶ Decision Center Business Console

As noted in 4.3.2, “Decision Center” on page 82, the Decision Center Enterprise Console is a more sophisticated environment that provides business users a significant level of control over rules. While it is not as powerful as Rule Designer, and hides the low-level implementation details, it supports the full range of rule management activities.

The Decision Center Business Console is a simpler environment for business users who are primarily focussed on maintaining rules and need to interact with others to track and discuss those changes with other users.

Deployment to Decision Center

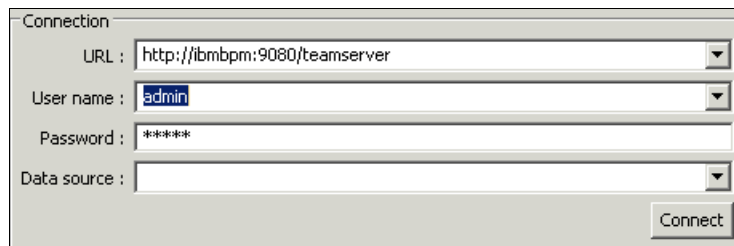
In order to examine the tools a business user could use to perform business management of rules, it is necessary to export the rule project and event project to Decision Center.

Note: The procedure below is identical for both Rule Designer and Event Designer. It will need to be performed in each IDE separately if you want to access each in the Decision Center. If you are already connected to Decision Center, you should disconnect. These instructions do not cover how to synchronize or delete an existing deployment.

1. Open or switch to the IDE from which you will perform the deployment (either Event Designer or Rule Designer).
2. Right-click the project (either **WarrantyClaimsRules** or **WarrantyEvents**) to be deployed and select **Decision Center → Connect**.

Note: If the deployment is being done from within Rule Designer, the project that is deployable to Decision Center is the Rule Project, not the RuleApp or the Java Project.

3. Enter the connection details for Decision Center. The address is the same as the address when Decision Center is opened from the Start Menu. The user name is typically defaulted to either admin or rtsAdmin and has a password that is the same as the user name. The parameters should be similar to those shown in Figure 4-32.



The screenshot shows a 'Connection' dialog box with the following fields and values:

- URL: http://ibmbpm:9080/teamserver
- User name: admin
- Password: *****
- Data source: (empty)

A 'Connect' button is located at the bottom right of the dialog.

Figure 4-32 Connection parameters for Decision Center

Note: The machine name and port number of your installation may be different. If you cannot connect, confirm that your server is running, and then try changing the machine name to localhost (assuming the server is running locally) and check the port it is running on; if port 9080 was already allocated when the server was installed, port 9081 may be used.

4. Click **Connect**.
5. Once the connection is established, click **Finish**.

Note: The next two steps may need to be done in reverse order, depending on how long you wait for the deployment to run.

6. Click **No** to avoid switching to the Synchronizing Perspective.
7. Click **OK** to accept the Synchronize Complete dialog.

Your project is now deployed in Decision Center.

Business user interface

It is now possible to log into either the Decision Center Enterprise Console or the Decision Center Business Console to view or modify the rules that were created. The sections that follow show how to access each interface and perform some simple navigation within it.

Decision Center Enterprise Console

To open the Decision Center Enterprise Console:

1. From the Windows desktop click **Start → IBM → WebSphere Operational Decision Management → Sample Server → Decision Center Enterprise Console**.
2. Enter the user name and password and click **Login**.

Note: The user name is typically defaulted to either admin or rtsAdmin and has a password that is the same as the user name.

You should see the home page of the Decision Center Enterprise Console. From here you can select which project you wish to access, as shown in Figure 4-33 on page 109.

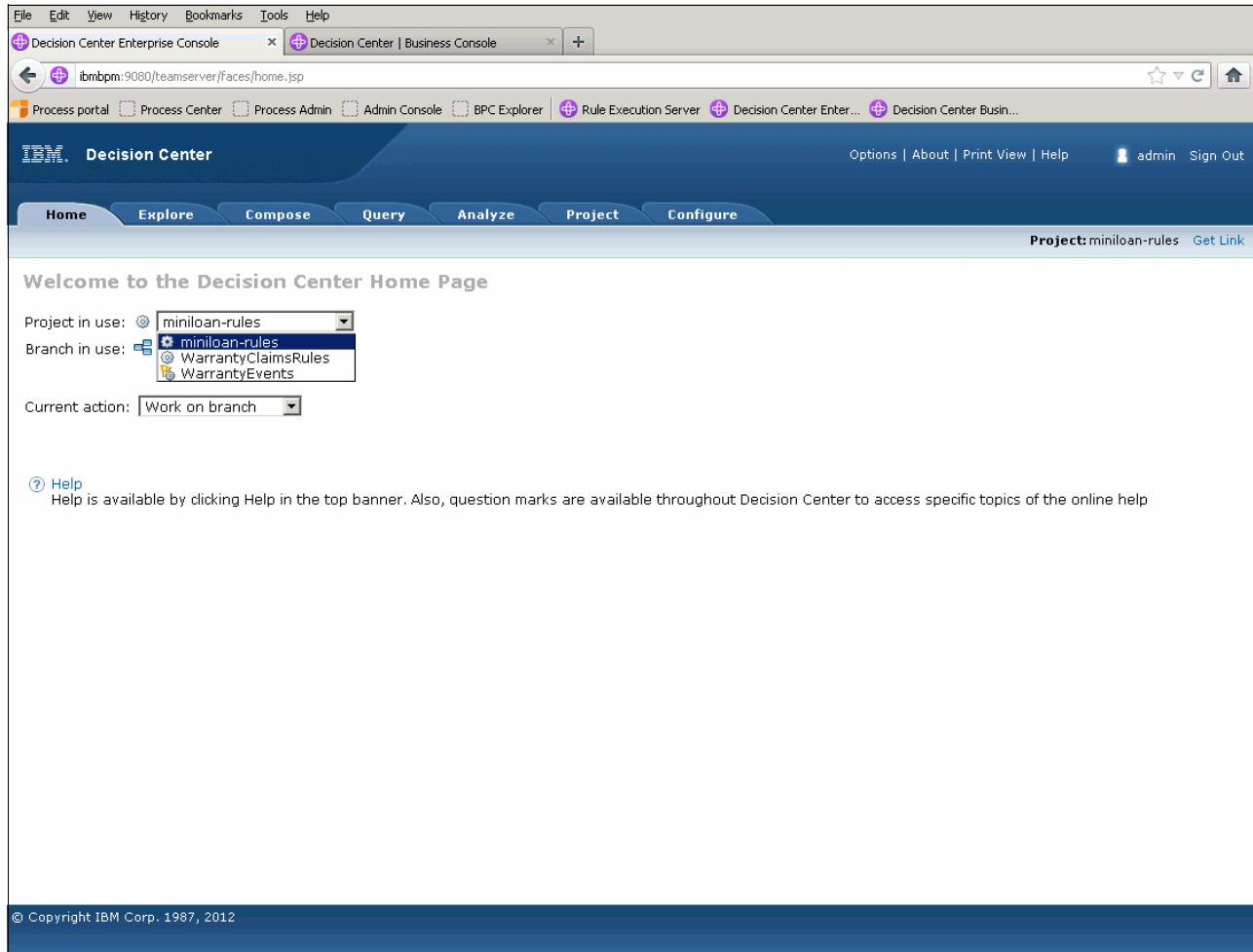


Figure 4-33 Decision Center Enterprise Console home page

You are now free to explore the project from a business user perspective. The Explore tab contains the project structure, and from here you can select rules and modify them. For example, to view the event rule that you created:

1. Set the Project in use in the home page to **WarrantyEvents**.
2. Click the **Explore** tab.
3. Click the **Rules** folder.
4. Click **MultipleWarrantyEvents**.

You should now see the rule as it appears in Figure 4-34.

```

Content
Event      EventReceived
Context    the serial number of the warrantyBusinessObject

if
  past occurrences of eventReceived within 3 hours is at least 1
then
  startWarrantyEventInBPM ;

```

Figure 4-34 A rule inside Decision Center Enterprise Console

Feel free to explore the interface and edit the rule.

Note: Editing the rule here will not change the rule in Event Designer. In order for the rules to be synchronized, this task must be performed from Event Designer and is beyond the scope of this chapter.

Decision Center Business Console

To open the Decision Center Business Console:

1. From the Windows desktop click **Start** → **IBM** → **WebSphere Operational Decision Management** → **Sample Server** → **Decision Center Business Console**.
2. Enter the user name and password and click **Login**.

Note: The user name is typically defaulted to either admin or rtsAdmin and has a password that is the same as the user name.

You can now see the What's New tab of the home page of the Decision Center Business Console. To view a rule perform the following:

1. Click the **Library** tab.
2. Click the **WarrantyClaimsRules** project.
3. Click the **automationStatus** rule.

You should see the rule that you created, and it should appear as shown in Figure 4-35.

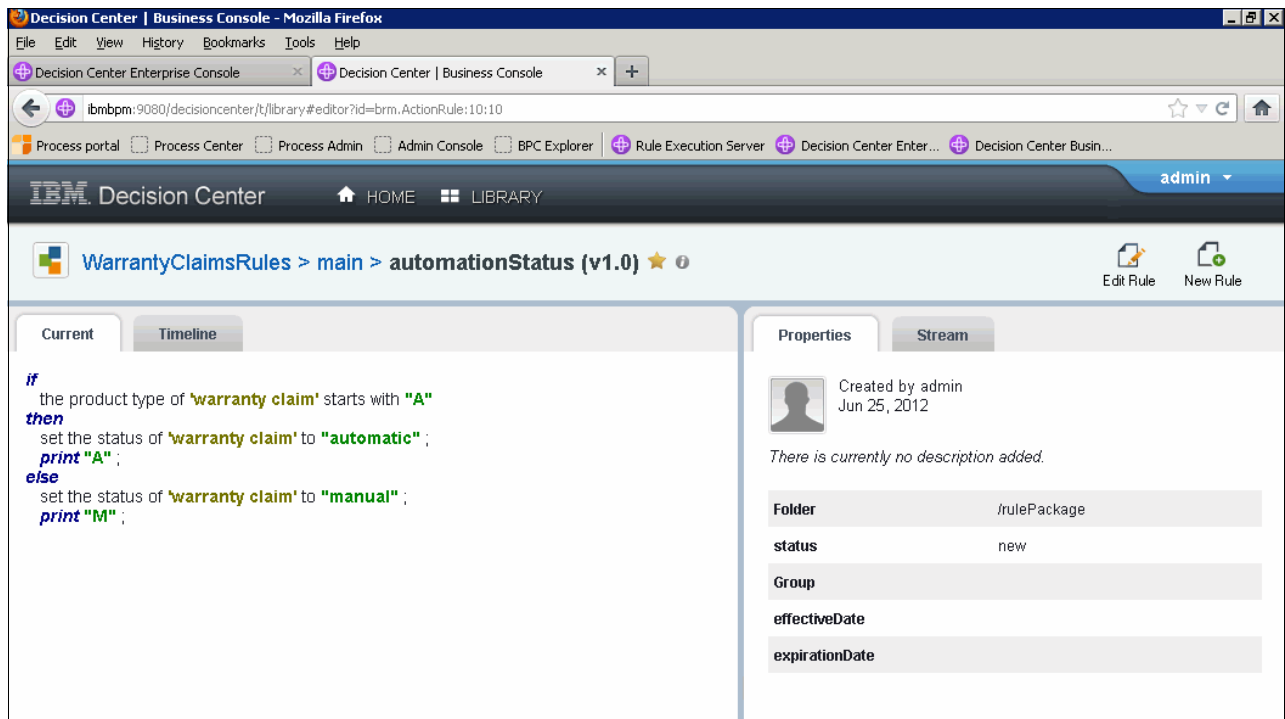


Figure 4-35 A rule inside Decision Center Business Console

This panel presents the rule on the left pane, while its properties are displayed in the right pane. If you click the **Timeline** tab in the left pane, you can see the history of this rule and view past versions. As there have not been any updates to the rule, this will not show anything of interest. If you click the **Stream** tab in the right pane, you will see all the comments and other activity that is associated with this rule. You can add your own comments here. Other

business users working on the same rule can use this to discuss and collaborate on any needed changes to the rule.



Business process management with Business Process Manager

Business process management (BPM) solutions enable an enterprise to choreograph processes and the process steps across disparate applications, people, and systems. Our warranty sample uses IBM Business Process Manager V8 to realize the advanced, integration-related components of this comprehensive BPM platform. We model the Warranty Business Process with it.

In this chapter we describe the following major components of our warranty sample:

- ▶ Warranty business process - Manages the core business process flow of the solution, and acts as the choreographer of process steps and activities.
- ▶ Integration with events and decision management - Manages the integration between process components and events, database, and decision management components.

Important: A basic knowledge of IBM Business Process Manager is needed to fully understand this chapter.

5.1 Overview of business process management

The notion of process optimization has been around for almost a century, as the concept started during the Industrial Revolution through “specialization.” Thought leaders worked on streamlining their processes for producing goods, be they physical or natural, in order to get more to markets at lower cost. After industrializing the world economy, this concept started to be adopted by business leaders wholeheartedly over the past decade, because they needed help to manage the interactions between system and humans. This discipline is business process management (BPM).

The key distinction between BPM and process optimization is its added focus on flexible and dynamic process design, process orchestration and automation through IT enablement. In addition to reduced cost through continued process improvement and automation, BPM provides the foundation for converged and agile business and IT responsiveness. Business leaders today look to it to help them get their goods and services to market, better, faster and cheaper than their competitors.

Figure 5-1 illustrates the core concepts of this discipline.

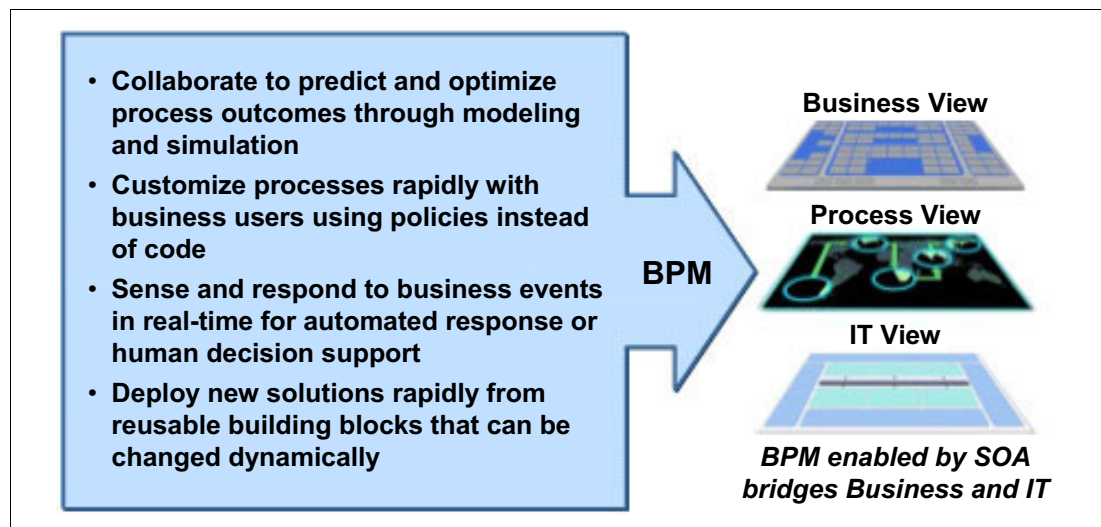


Figure 5-1 BPM drives business and IT alignment and responsiveness

Intrinsic to BPM is the principle of continuous operational improvement, perpetually increasing value generation and sustaining market competitiveness, or dominance. BPM focuses on driving overall bottom-line success by integrating business verticals and optimizing core work (for example order-to-cash, integrated product development, and integrated supply chain). This focus helps direct the deployment of resources throughout the organization into efficient processes that create customer value—which differentiates BPM from traditional (that is, compartmentalized) functional management disciplines.

5.1.1 Managed business process

A managed business process is a process in which stakeholders and process owners have both visibility into the process and the ability to modify it in order to produce better business outcomes. With process control, you can make informed decisions on how best to change the behavior of your process. These decisions might affect the process itself, or they may impact domains outside of the business.

As a manager, you regularly participate in decisions for change (Figure 5-2). All management decisions, at any level in the organization, can be associated with one or more of the following domains: corporate strategy, business resources, and business processes.



Figure 5-2 Domains of change for a managed business process

Examples of corporate strategy decisions include entering new markets, discontinuing a product, and selling assets. You can imagine how these decisions might impact or lead to decisions in the business resource domain such as hiring or training new human resources, outsourcing jobs, investing in technology, or making capital improvements to facilities. Decisions in the business process domain might include changing a decision threshold in a process flow to reduce the human workload (that is, lower the threshold) or reducing risk (raising the threshold) by changing the behavior of the process. Business process decisions might also include feedback to change the process by adding or removing activities.

5.1.2 Process automation, visibility, and control

Process automation, visibility, and control are compounding elements of the business impact realized by BPM (see Figure 5-3). Process automation immediately accrues business value by increasing efficiency, reducing errors, eliminating process variation, and removing rework for human tasks. It is important to recognize, however, that automation is not the end goal of process improvement via BPM.

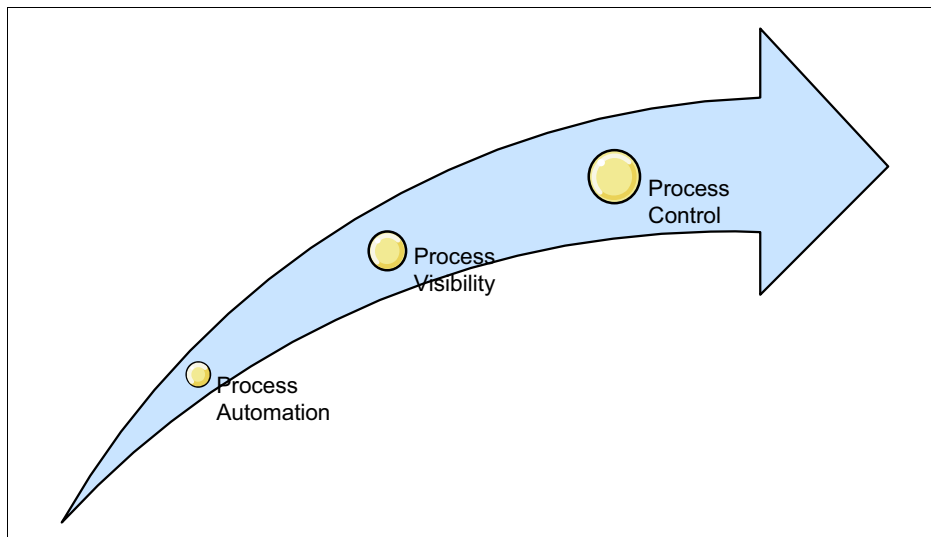


Figure 5-3 Elements of compounding values of a managed process

Process visibility allows you to see new aspects of your processes in tangible ways, and in some cases, in real time (when invoked). These capabilities provide insight into the performance of your processes, which can help identify areas for improvement through timely action. In addition, all get a full, end-to-end view through process visibility.

Process control, on the other hand, is what ultimately differentiates a managed process from unmanaged ones. Having control over your design-time and runtime business processes means you can engage the right skills to address problems or effect change in a timely manner. By default, control demands full governance of your processes to ensure all are operating consistently and in compliance with both internal and external policies and regulations.

5.2 Overview of IBM Business Process Manager

IBM Business Process Manager is a comprehensive and consumable business process management platform that provides visibility and management of your business processes. It allows IT to enable business users and managers to track their entire business operation on a single dashboard, receive alerts, and subsequently drill down to the lowest level of instance detail.

Figure 5-4 on page 117 depicts the major components of IBM Business Process Manager.

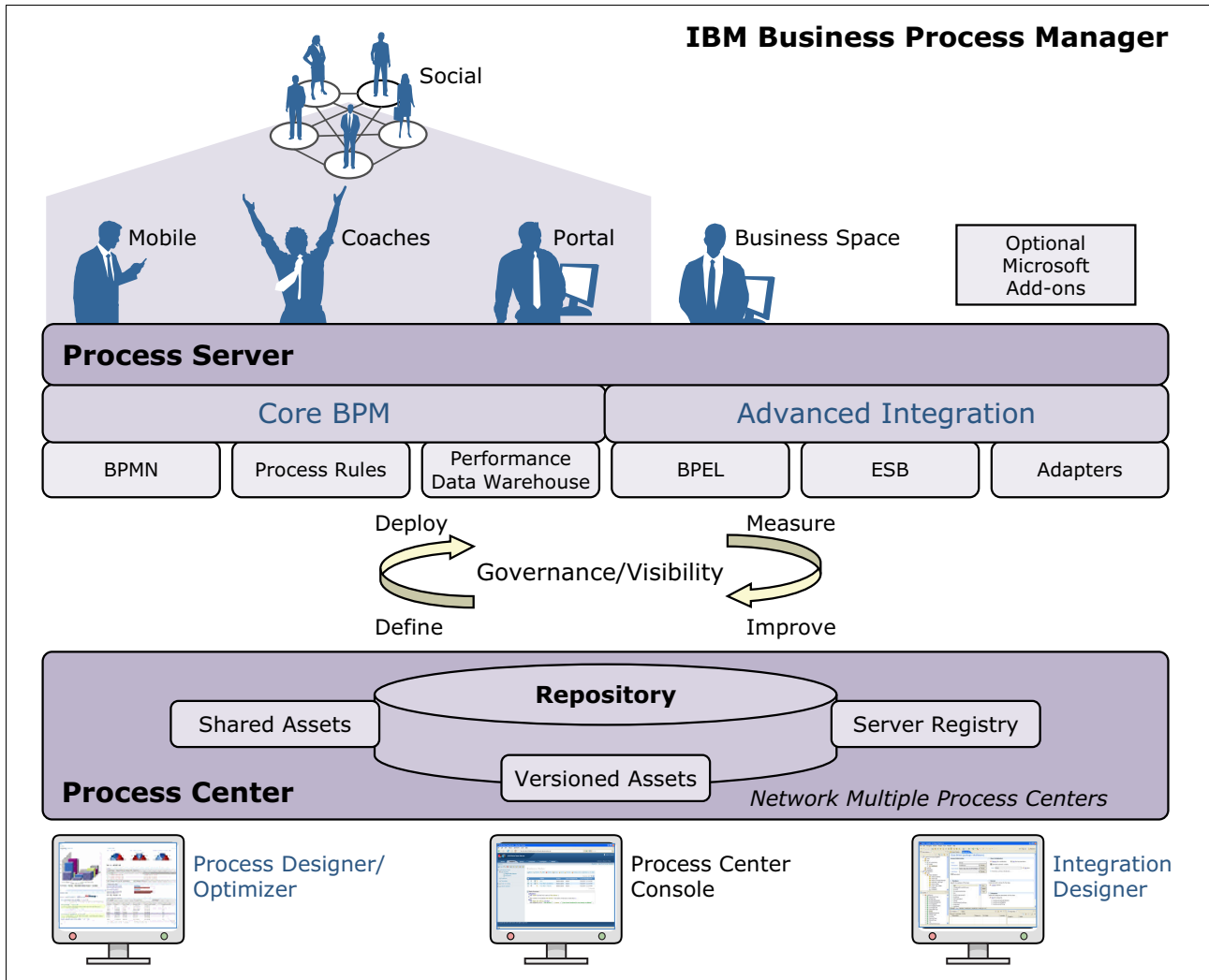


Figure 5-4 Major components of IBM Business Process Manager

In the following sections we describe the available configurations of IBM Business Process Manager.

5.2.1 IBM Business Process Manager configurations

IBM Business Process Manager V8.0 is available in the configurations listed in Table 5-1.

Table 5-1 IBM Business Process Manager configurations

Configuration	Phase
Advanced	<p>Transformation</p> <p>Complete set of business process management capabilities:</p> <ul style="list-style-type: none"> ▶ Extended support for high-volume process automation ▶ Built-in SOA components for extensive enterprise-wide service integration and orchestration

Configuration	Phase
Standard	Program Configured for typical business process management projects: <ul style="list-style-type: none"> ▶ For multiproject improvement programs, with high business involvement ▶ Basic system integration support ▶ Rapid time-to-value and improved user productivity
Express	Project Configured for first business process management project: <ul style="list-style-type: none"> ▶ Rapid time-to-value: improved user productivity ▶ Low entry price ▶ Easy installation and configuration

IBM Business Process Manager is a single BPM platform that combines human-centric and integration-centric capabilities into a unified product, which can be consumed per the capability requirements shown in Table 5-2.

Table 5-2 IBM Business Process Manager configuration capabilities

Capability	Advanced	Standard	Express
WebSphere Lombardi Edition compatible execution	X	X	X
IBM Process Designer (BPMN)	X	X	X
Collaborative editing and immediate playback	X	X	X
Interactive “process coach” user interfaces	X	X	X
ILOG-based process rules	X	X	X
Process Portal	X	X	X
Real-time monitoring and reporting	X	X	X
Performance analytics and optimizer	X	X	X
Performance Data Warehouse	X	X	X
Process Center and shared asset repository	X	X	X
Unlimited process authors and users	X	X	200 users and 3 authors
High availability: clustering and unlimited cores	X	X	<ul style="list-style-type: none"> ▶ 4 cores production ▶ 2 cores development ▶ No cluster
WebSphere Process Server compatible execution	X		
Integration Designer (BPEL and SOA)	X		
Built-in enterprise service bus (ESB)	X		
Transaction support	X		
Integration adapters	X		
Flexible Business Space user interface	X		

Capability	Advanced	Standard	Express
Advanced platform support (Linux on System z®, IBM AIX®, Solaris)	X	X	

5.2.2 Process Center

Process Center is a runtime environment that includes a repository for all process models, services, and other assets created via IBM Process Designer or IBM Integration Designer, the IBM Business Process Manager authoring environments. It has a console with the tooling you need to maintain this asset repository.

From the Process Center console administrators can:

- ▶ Install process applications that are ready for testing or production on the process servers in those environments.
- ▶ Manage running instances of process applications in configured environments.
- ▶ Grant appropriate authorization for users and groups to access the repository.

Administrators who do not actively work in the Designer view can use the Process Center console to provide a framework in which BPM analysts and developers can build their processes and underlying implementations.

5.2.3 Process Server

Process Server provides a single BPM runtime environment that can support a range of business processes, service orchestration, and integration capabilities. Since it is integrated with Process Center, you can run your processes as you build them directly in the authoring environments.

When you are ready, you can install and run those same processes on the process servers in your runtime environments to test and prepare each for full deployment. The Business Performance Data Warehouse component collects and aggregates process data from your processes running on the process servers, which enables you to test and refine your processes before formal deployment. You can use this data to improve your business processes.

5.2.4 Authoring environments

IBM Business Process Manager Advanced offers two authoring environments:

- ▶ IBM Process Designer is used to model business processes that involve human tasks.
- ▶ IBM Integration Designer is used to build services that are self-contained or that invoke other existing services such as web services, enterprise resource applications, or applications running in CICS® and IMS™.

Normally, a business analyst creates a new process in IBM Process Designer and defines an advanced integration service that the process would use. An integration developer creates that service in IBM Integration Designer and publishes it so the business analyst can use it to complete the process. The advantage from using these authoring environments is that both the business analyst and the integration developer can collaborate effectively via the Process Center repository, which reduces the time to build, test and deploy new or revised business processes.

IBM Process Designer

IBM Process Designer is available in all IBM Business Process Manager editions, and it comes with easy-to-use, graphics-oriented tooling for you to create process models, reports, and simple services. In addition, you can call a service created in Integration Designer via an interface to access back-end systems or obtain customer data. In short, IBM Process Designer focuses on the business process and IBM Integration Designer focuses on automated services to complement the business process.

Process applications developed in IBM Process Designer can at any time be run on the Process Center server or saved to a snapshot and deployed on the Process Server. The same is true of services developed in Integration Designer and associated with process applications.

Integration Designer

Integration Designer is available in IBM Business Process Manager Advanced Edition or as a stand-alone toolset. It is designed as a complete integration development environment for those building integrated applications. Integration developers use it to call applications on Enterprise Information Systems (EIS), involve business processes across departments or enterprises, and invoke applications locally or remotely written in a variety of languages and running on a variety of operating systems.

Integration Designer tools are based on a service-oriented architecture. All services created with this tooling comply to the leading, industry-wide standards. Through the use of visual editors that abstract service components from their implementations, integration developers can assemble integrated applications without having detailed knowledge of the underlying implementation of the components.

In Integration Designer, components are assembled in modules. Imports and exports are used to share data between modules. Artifacts placed in a library can be shared among modules. Modules and libraries can be associated with a process application for use with the Process Center and can be used as services by processes created in IBM Process Designer.

In such cases, processes can also be deployed with the process application. Alternatively, modules and libraries can be deployed directly to the test environment or to the Process Server. You can use mediation modules to create mediation flows, which you can deploy to WebSphere Enterprise Service Bus or to Process Server.

5.2.5 Administration tools

IBM Business Process Manager includes a set of administration tools to help you accomplish tasks ranging from installing and managing snapshots to administering processes and working with the resources in your IT environment.

Command-line tools

The command-line tools, scripting interfaces, and programming interfaces enable you to administer your runtime environment.

- ▶ Command-line tools are simple programs that you run from an operating system command-line prompt to perform specific tasks. Using these tools, you can start and stop application servers, check server status, add or remove nodes, and so on.
- ▶ The WebSphere administrative (wsadmin) scripting program is a nongraphical command interpreter environment that enables you to run administrative options in a scripting language and to submit scripting language programs for execution. It supports the same

tasks as the administrative console, as well as many of the Process Center console tasks. The wsadmin tool is intended for production environments and unattended operations.

- ▶ Administrative programming interfaces are a set of Java classes and methods under the Java Management Extensions (JMX) specification that provide support for administering Service Component Architecture (SCA) and business objects. Each programming interface includes a description of its purpose, an example that demonstrates how to use the interface or class, and references to the individual method descriptions.

The Process Center console

The Process Center console provides a convenient location for users to create and maintain high-level library items such as process applications and toolkits. It helps provide a framework in which BPM analysts and developers can build their processes and underlying implementations. In addition, the Process Center console provides tools for maintaining the repository, including setting up the appropriate authorization for users and groups.

You can access the Process Center console through a web browser:

<http://host:9080/ProcessCenter>

The Process Admin console

The Process Admin console is used to administer the process servers in your environment, including the users and installed snapshots for each server. In addition, it provides tools to help you manage queues and caches.

The Process Admin console includes the Process Inspector, a tool to view and manage process instances for process applications that are running on a specific process server.

You can access the Process Admin console through a web browser:

<http://host:9080/ProcessAdmin>

The Business Performance Admin console

The Business Performance Admin console includes tools for managing the Performance Data Warehouses in your environment. You can use this tool to manage server queues and monitor server performance.

You can access the Business Performance Admin console through a web browser:

<http://host:9080/PerformanceAdmin>

The WebSphere Application Server administrative console

The administrative console is used to administer applications, services, and other resources at a cell, node, server, or cluster scope. You can use the console with stand-alone servers and with deployment managers that manage all servers in a cell in a networked environment.

If you have installed a stand-alone profile, you have a single node in its own administrative domain, known as a cell. Use the administrative console to manage applications, buses, servers, and resources within that administrative domain. Similarly, if you have installed and configured a network deployment cell, you have a deployment manager node and one or more managed nodes in the same cell. Use the administrative console to manage applications, set up managed nodes in the cell, and monitor and control those nodes and their resources.

You can access this console through a web browser:

<http://host:9043/ibm/console>

Business Process Choreographer Explorer and Business Process Archive Explorer

Depending on your user role, you can use these client interfaces to manage BPEL processes and human tasks created in IBM Integration Designer, work with your assigned tasks, view completed BPEL processes and human tasks that are in an archive database, or delete processes and tasks from the archive.

Business Space powered by WebSphere

Business Space powered by WebSphere is an integrated user experience for business users across the IBM business process management portfolio. Business Space provides a customizable and collaborative environment for you to monitor, review, and administer common business processes, such as human task flows, modeling, and performance indicators.

Business Space is a browser-based graphical user interface that you can use to view and interact with content from various products in the business process management portfolio. Business Space not only provides a single web-based point of access for the content, you can use Business Space to combine the content in useful and interesting ways. These combinations can give you insight into your business and the capability to react to changes.

Business process rules manager

The business process rules manager is a web-based tool that assists the business analyst in browsing and modifying business rule values. The tool is an option of IBM Process Server that you can select to install at profile creation time or after installing the server.

5.3 Creating the Warranty Sample

Now that we have discussed the components and merits of IBM Business Process Manager, we use it to create a business process for the warranty sample application, which is designed to process warranty claim exceptions. These exceptions are defined as multiple claims coming in for the same item (that is, identified by its serial number) over a short period of time.

In our example, a business analyst is assigned the responsibility for creating the Warranty Exception Process using IBM Process Designer, while an integration developer is tasked to create the integration of the events and rules with the business process through the mediation layer.

To create the process:

1. Log in to IBM Process Designer.
2. Create a new process application by clicking the link **Create New Process App** as indicated by the red arrow in Figure 5-5 on page 123.

This process application acts as a container or package for business processes and related artifacts.
3. Enter the name of the new application as Redbook Warranty Processes, provide an acronym, for example RWP, and enter a short description for your new process application.

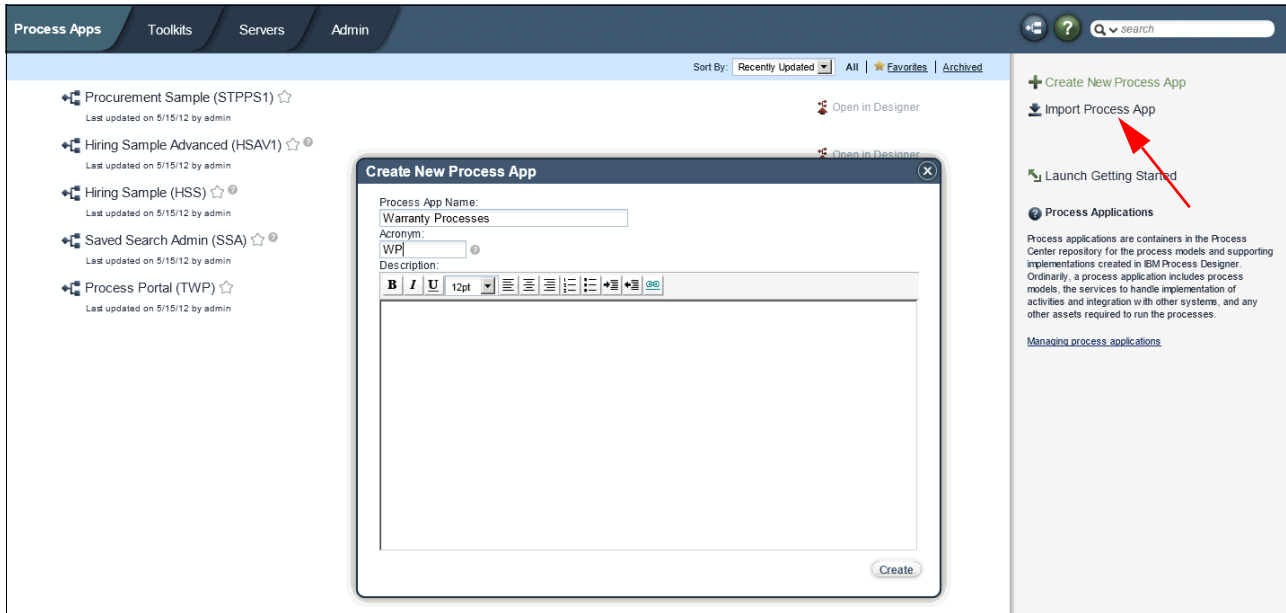


Figure 5-5 Create a new process application

4. Open Warranty Processes' process application in Designer using the **Open in Designer** link as indicated by the red arrow in Figure 5-6.

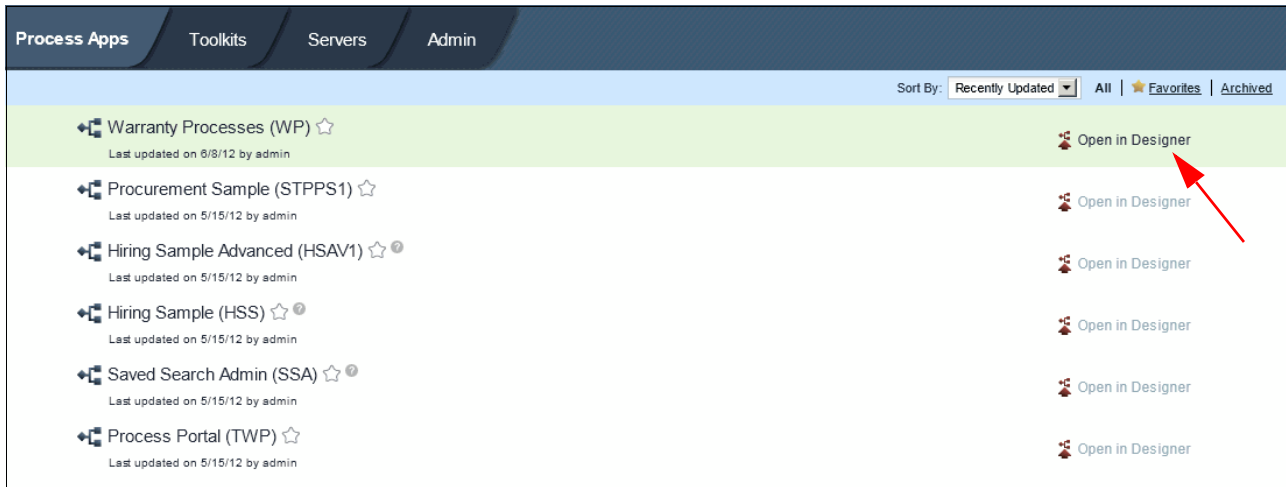


Figure 5-6 Open the newly created process application

5. Create a new business process definition by selecting **Processes** → **Business Process Definition** (click the + sign beside Processes) as shown Figure 5-7 on page 124.
 To model a process in IBM Process Designer, create a *business process definition* (BPD), which is a reusable model of a process that defines what is common to all runtime instances of that process model.

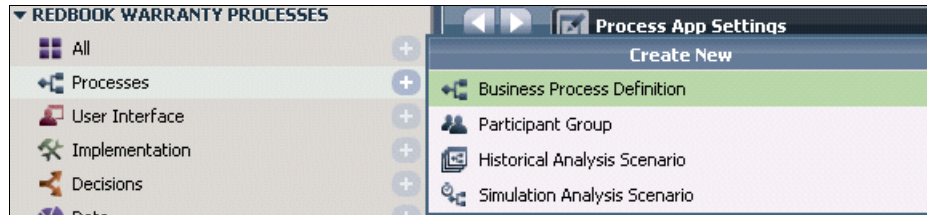


Figure 5-7 Create a new BPD

- Next, name your new BPD Warranty Exception Process, and click **Finish** to proceed (see Figure 5-8).

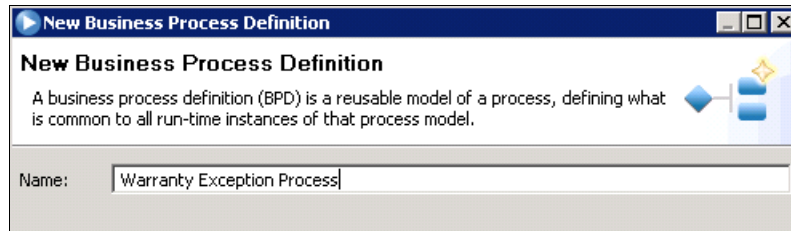


Figure 5-8 Name the new BPD

- IBM Process Designer generates your process diagram with the initial modeling constructs as shown in Figure 5-9.

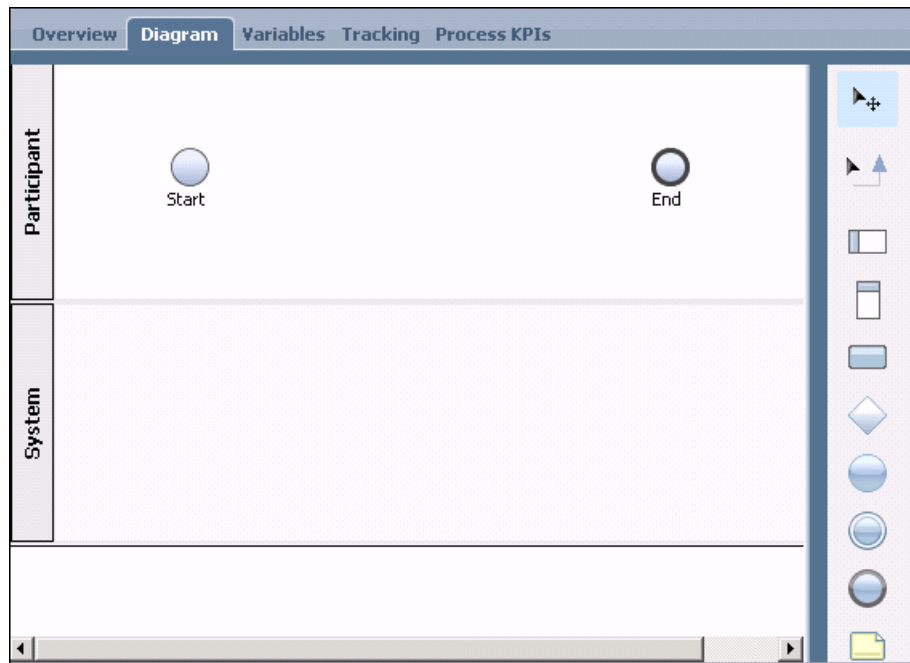


Figure 5-9 Default process diagram

A BPD must contain a start event, an end event, at least one swim lane, and one or more activities, which is why the default process diagram contains these initial modeling constructs.

A swim lane is a visually separated row within a process flow diagram that groups all the activities in the process that are performed by a particular combination of roles, resources,

organization units, or locations. A BPD needs to include a swim lane for each system (a system lane) or group of users who participate in a process (a participant lane).

You can designate any specific person or group to be responsible for the activities in a participant lane. Each lane that you create is assigned to the All Users participant group by default. You can use this default participant group for running and testing your BPD in the Inspector. The All Users participant group includes all users who are members of the tw_allusers security group, which is a special security group that automatically includes all users in the system.

A system lane contains activities handled by a specific IBM Process Center system. Each activity needs an implementation, which defines the activity and sets the properties for the task. During implementation, a developer creates a service or writes the JavaScript necessary to complete the activities in a system lane.

For the Warranty Exception Process, you need to create the following lanes:

- Claim Officer
- Claim Manager

Note: For explicit guidance on how to add and adjust swim lanes, we refer you to:

http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/adding_swim_lanes.html

8. When done, your diagram should look like Figure 5-10.

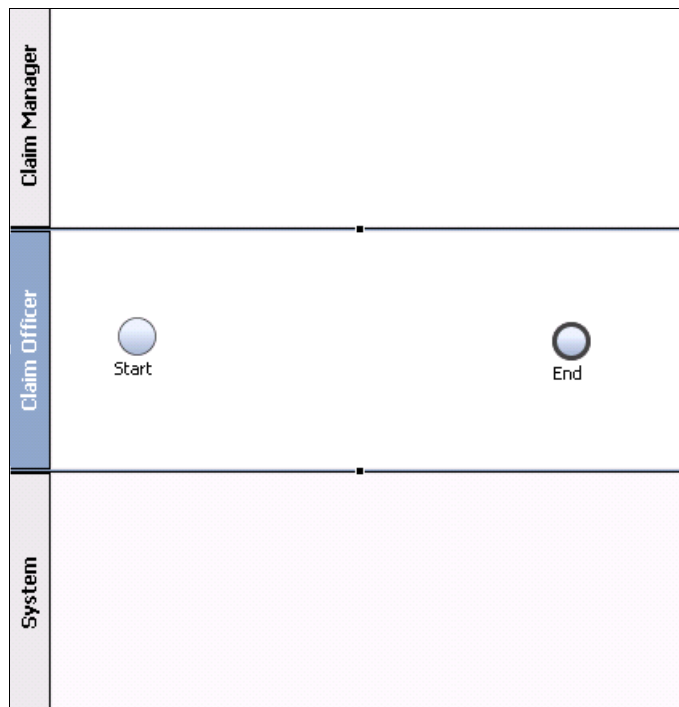


Figure 5-10 Warranty Exception Process Diagram

9. Next, you need to model your business process, which entails the following high-level steps:
- Dragging modeling constructs from your palette onto the diagram area
 - Specifying the details for a modeling construct by selecting it in your diagram and editing the properties in the Properties view.

Figure 5-11 illustrates the activities to be added for Warranty Exception Process. The activities in the diagram use the following color convention. It is not mandatory to do so, but any such convention used can be useful for later readers to understand the workflow.

- Blue for Human task activities
- Red for Rule usage activities
- Green for Database activities

Note: For explicit guidance on how to model processes, we refer you to:
http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/establishing_process_flow.html

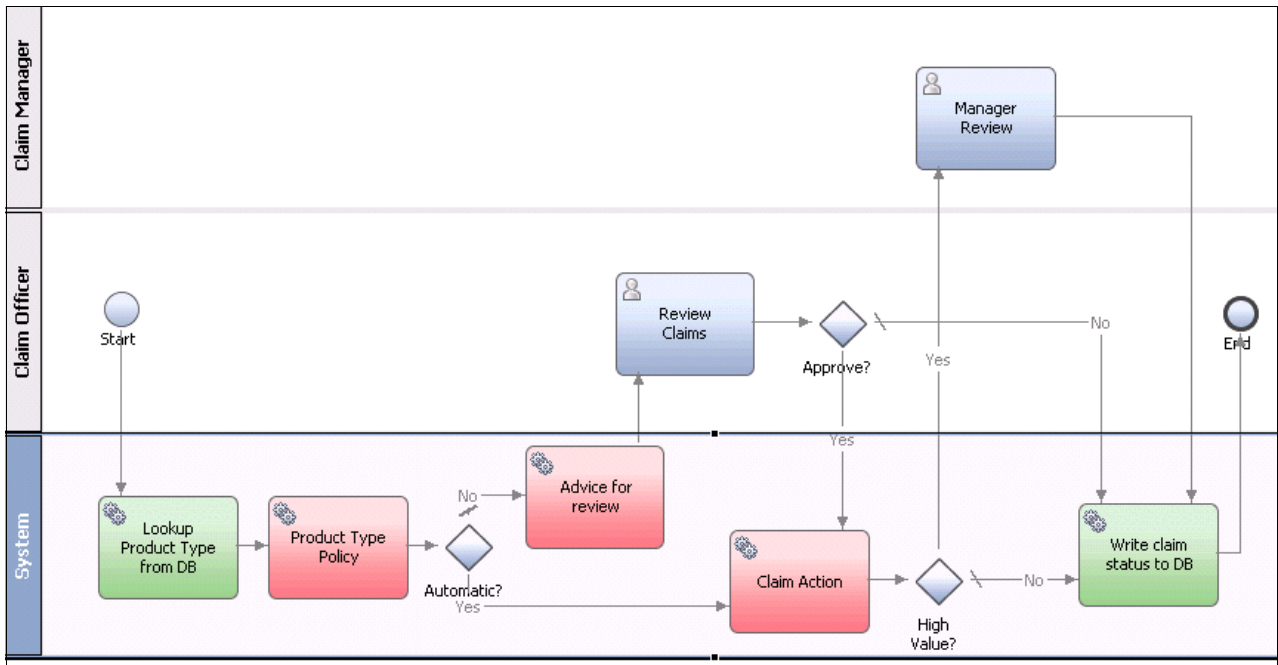


Figure 5-11 Warranty Exception Process diagram

Table 5-3 describes each of the activity steps of the Warranty Exception Process. Use the information shown in Table 5-3 to create the process diagram shown in Figure 5-11.

Each activity step in the business process needs to be assigned a *task type* and the implementing *service*. Table 5-3 lists all the task types and the service types that are available in IBM Business Process Manager.

Table 5-3 Warranty Exception Process tasks and services

Steps/Activities	Description	Task Type	Service Type
Start Event	Trigger for the process to start.	N/A	N/A
Lookup Product Type from DB	Automated database lookup - Using the product serial number, retrieve the product type from the database.	System Task	Integration Service
Product Type Policy Rules	Execute rules - Based on product type, execute rules to determine whether the claim exception process should proceed automatically or needs manual intervention.	System Task	Decision Service

Steps/Activities	Description	Task Type	Service Type
Decision Gateway	Split execution path based on the result of the previous step.	N/A	N/A
Advice for Review	Execute rules - Based on parameters such as period of time between claims, determine the advice to be given to the Claim Officer.	System Task	Decision Service
Review Claims	Claim Officer reviews the claim along with the advice provided by the system through automated rule execution.	User Task	Human Service
Decision Gateway	Split execution path based on the result of the review by Claim Officer, that is, whether or not it is approved.	N/A	N/A
Claim Action	Execute rules - Based on parameters such as product type, problem type, and claim value, execute rules to determine if the claim requires additional management review.	System Task	Decision Service
Decision Gateway	Split the execution path based on the result of the previous step where the rules are executed.	N/A	N/A
Manager Review	A Claim Manager must review any claim where the value (as determined by Claim Action) is high, for example if the action is to replace a high value part with a new one.	User Task	Human Service
Write to Warranty Claim DB	All claims (both approved and rejected) are written to an external database with the status of the claim.	System Task	Integration Service

5.4 Add rule invocation for rules-related activities

Note: In this section we assume that you have first completed 4.4, “Overview of business rules” on page 84. In this section, we create a Decision Service that uses the Business Rules application (WarrantyClaimsRulesApp) described in 4.4, “Overview of business rules” on page 84.

Our Warranty Exception Process contains activities with a Decision Service. This type of service is used when you want a decision or condition in a business rule to determine which process implementation is invoked.

5.4.1 Create a new Decision Service

To create a new Decision Service:

1. Open IBM Process Designer; if you already have IBM Process Designer open, ensure that you are in the Designer View.
2. On the top left corner of the window, click the plus symbol to the right of Decisions, and then select Decision Service, as shown in Figure 5-12 on page 128.



Figure 5-12 Create new Decision Service

3. Name the service AutomateBasedOnProductType as shown in Figure 5-13, and click **Finish** to proceed.

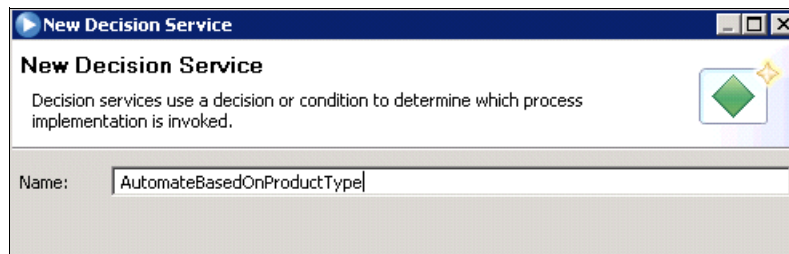


Figure 5-13 Create a new Decision Service

A Decision Service contains one or more of the following components:

- ▶ **BAL Rule**

You can use the rule editor in this component to author business rules using Business Action Language (BAL), which is a declarative language that relates business concepts to business data and actions.

- ▶ **JRules Decision Service**

IBM Business Process Manager integrates with IBM WebSphere ILOG JRules and WebSphere Operational Decision Management Server using the JRules Decision Service component. You can use this rule component to connect to and implement rule applications that are available on a JRules Rule Execution Server.

- ▶ **Decision Table**

The Decision Table component contains a rule table. Each row in the rule table represents a Boolean condition that evaluates to true or false at runtime. When a rule evaluates to true, the JavaScript expression that you provide as the rule action is started.

For our Warranty Exception Process we have chosen to use JRules Decision Service to create our Decision Service so that we can connect to the WebSphere Operational Decision Management Server to use the Business Rules application we created in Section 4.4, “Overview of business rules” on page 84.

When building a Decision Service, follow these guidelines:

- ▶ Build your rule hierarchy so that rule conditions are ordered from most complex to least complex.
- ▶ Create a final condition that is a catch-all rule. This rule is necessary if you cannot guarantee that the variable you want to modify in a rule will be set before running the process that triggers your Decision Service.

- ▶ Consider encapsulating your rules in a single-function Decision Service, one that allows the service to be available to any other part of the process application that may need the same rule logic.

Note: For explicit guidance on building a Decision Service, we refer you to:

http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/topics/crule_ruleserv_ovw.html

5.4.2 Connect with WebSphere Operational Decision Management

To connect with the WebSphere Operational Decision Management server:

1. Double-click the Decision Service **AutomatedBasedOnProductType** that you created in Section 5.4.1, “Create a new Decision Service” on page 127.
2. After the service diagram opens, drag a JRules Decision Service from the palette onto your diagram and name it *JRules Product Type automation*.
3. Select the arrow tool from the palette, link the start node to the newly created Decision Service, and then link that service to the end node.
4. With the JRules Decision Service component selected, click the Implementation option in the Properties tab.
5. In the Discovery section, enter the Server, SOAP Port (if the Rule Execution Server is running on WebSphere Application Server) and user name and password (in case of a secure connection) to connect to a Rule Execution Server that contains deployed rule applications (Rule Apps) that you want to use.

Tip: The SOAP port, user name, and password fields accept embedded JavaScript expressions, so you can use variables to provide those values.

6. Select the **Implementation** option in the Properties tab (Figure 5-14) followed by the **RulesServer** from the server drop-down menu.

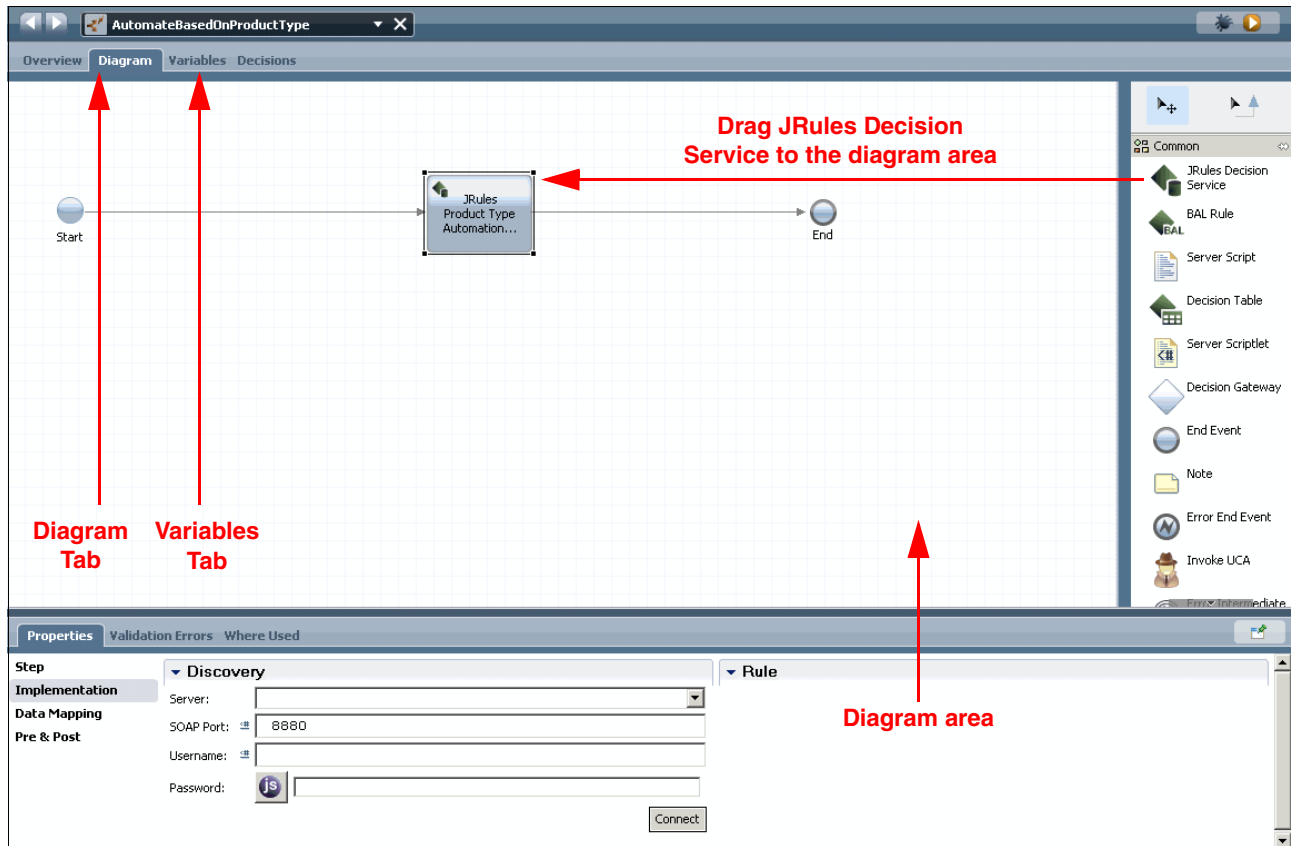


Figure 5-14 Create a new Decision Service

Note: If your Decision Server is not started, you need to start the same server that you used to build the service.

7. Next, enter the user name and password that will authenticate to your Decision Server instance, and click **Connect**.

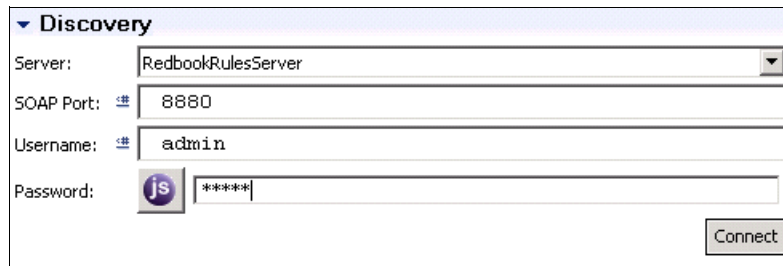


Figure 5-15 Discover the Rules Application to link with Decision Service

8. In the Rule section, select the Rule App that you want followed by the version that you want to use. For the Warranty Exception Process, use the WarrantyClaimsRuleApp Rule App and WarrantyClaimsRules Ruleset, as shown in Figure 5-16 on page 131.

Important: If a secure connection to the Rule Execution Server has not been established, the menu is not populated. In this case, manually enter the name and version of the Rule App and Ruleset that you want to use. The names must be accurate for the next step to work.

The screenshot shows a dialog box titled "Rule" with a dropdown arrow on the left. It contains two rows of configuration. The first row is "Rule App:" with a text box containing "WarrantyClaimsRuleApp" and a dropdown arrow, followed by "Version:" with a dropdown menu showing "Latest". The second row is "Ruleset:" with a text box containing "WarrantyClaimsRules" and a dropdown arrow, followed by "Version:" with a dropdown menu showing "Latest". Below the "Rule App:" field is a "View" button. Below the "Ruleset:" field is a "Generate Types..." button.

Figure 5-16 Discover the Rules Application to link with Decision Service

9. Click **Generate Types**.
10. In the Generating Types Wizard, make sure the “Generate request/response wrapper types” option is *not* selected (Figure 5-17) before clicking **Next**.

The screenshot shows a dialog box titled "Generate Types Wizard" with a blue header bar. Below the title bar is the text "Generating Types" and "Click Next to begin type generation". There is a checkbox labeled "Generate all types" which is unchecked. Below this is the text "Select operation(s) for which you want types to be generated:" followed by a list box containing "WarrantyClaimsRules". At the bottom of the dialog is another checkbox labeled "Generate request/response wrapper types" which is also unchecked. At the very bottom are four buttons: "< Back", "Next >", "Finish", and "Cancel".

Figure 5-17 Generate data types

11. Click **Next** through the remaining wizard windows until the type generation is complete, and then click **Finish** to accept default type generation (Figure 5-18).

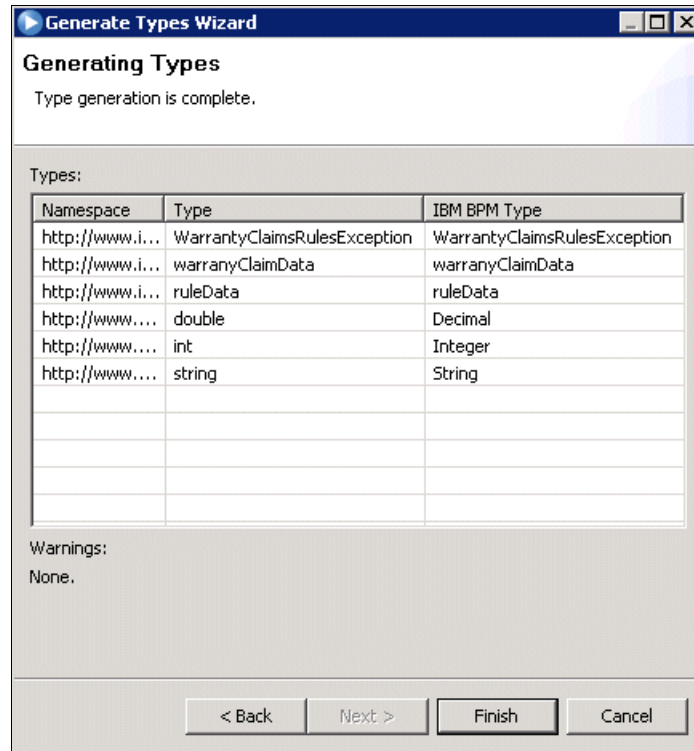


Figure 5-18 Generate data types

After verifying that your business objects have been created, examine them to check that they match with the parameters you defined when creating your rule project:

1. Go to the Variables tab (Figure 5-14 on page 130) and click **Add Input**.
2. Specify the name of the input data type as `input` and the type as `ruleData`, and select the **Has Default** option.
3. Click **Add Output**, and specify the name of the output data type as `output` and the type as `ruleData`.
4. Click the Diagram tab (Figure 5-14 on page 130).
5. Click **back** on the JRules Rule Service that you created.
6. Select the Data Mapping section (Figure 5-19 on page 133) and enter a value of 0 for the Decision ID.

Note: this value can be used for correlating business processes and decisions by passing the process instances ID as a parameter. This scenario does not use it for any purpose, but it is a mandatory field. A value must be provided.

7. Map the `tw.local.input` input type to `ruleData`.
8. Map the `tw.local.output` output type to `ruleData`.

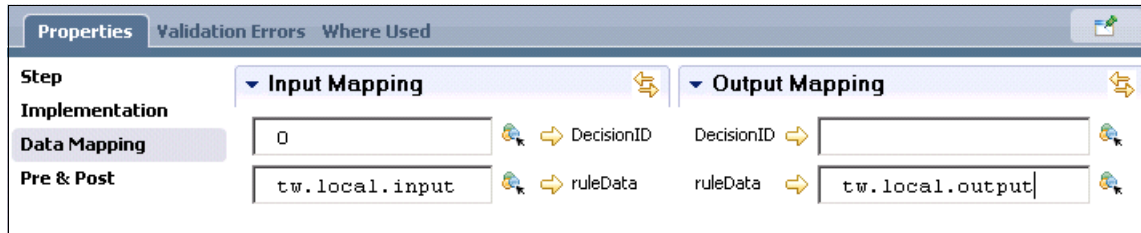


Figure 5-19 Data Mapping

Note: For explicit guidance on adding a JRules Decision Service component to a service, we refer you to:

http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/topics/building_rule_service_C.html

5.4.3 Test the Decision Service

To test a rule component and the rules it contains, you can load the Decision Service with default data and then step through the activities in your BPD to see the generated process data as it interacts with your defined business rules. For example, if you set a breakpoint on an activity that has an associated Decision Service, you can make sure that the Decision Service is producing the data that you expect, and that it is not producing any error messages or exceptions.

To create a test process to execute the rules:

1. Click the plus symbol next to Processes and choose to create a new BPD.
2. Name the BPD `TestProductTypeRule` and then click **Finish**.
3. Click **Decisions** and drag the `AutomateBasedOnProductType` service into the system swim lane of the process diagram.
4. Connect the start node to the Decision Service using a connection arrow from the palette.
5. Next, connect the Decision Service to the end node in similar fashion as step 4.
6. Go to the Variables tab, click **Add Input**, and specify the name of the input data type as `input` and the type as `ruleData`.
7. Then select the **Has Default** option and enter the value `Apple` in the `productType` field.
8. Click **Add Output** and specify the name of the output data type as `output` and the type as `ruleData`.
9. Click the diagram tab and select the **AutomateBasedOnProductType** service.
10. Go to the Data Mapping tab, set the input mapping to `tw.local.input` and the output mapping to `tw.local.output`.
11. Now your process and rule are ready to test; click **Play**, and if asked to switch the view to the Inspector view, click **Yes**.

Note: You will see your process execute in the Inspector view. Wait until the process and Decision Service execute. The first time your process executes, all the service artifacts are compiled, so it may take between 5 and 30 seconds to complete. Refresh the view every few seconds, and do not proceed until the status of your activity is complete and the process instance status is closed.

12. In the Execution State tab, select **TestProductTypeRule** and then click the **Variables** tab to view the input and output parameters.
13. Select the input parameter to verify that the XML document contains the Apple input parameter that you set.
14. Click the output parameter to verify that the status type is set to automatic in the XML document.

Note: For additional testing guidance, we refer you to:

http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/topics/trule_service_testing.html

5.5 Add the user interface for human task activities

Users interact with, and participate in, a business process through user interfaces. Human services (*human tasks*) represent the actions that these users perform within the process, and *coaches* are the user interface for these services. In an IBM Business Process Manager process, users interact with services and the service data using coaches. Coaches feature reusable *coach views*, which are reusable user interfaces that you can create and customize. Coach views consist of one or more other coach views, data bindings, layout information, and behavior.

To create a new human service and add the appropriate coaches to the user interface for this service:

1. First, create a new human service by clicking **Human Service**, as shown in Figure 5-20.



Figure 5-20 Create new human service

2. Enter a name for the service (Figure 5-21) and click **Finish**.

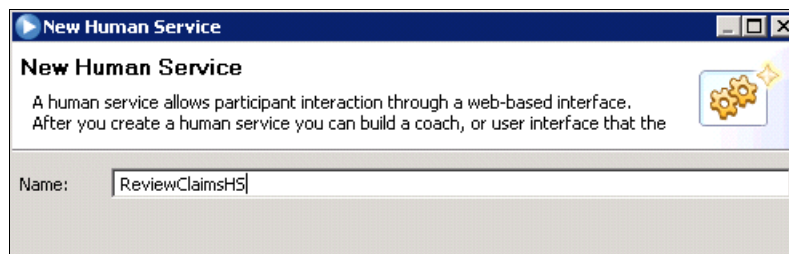


Figure 5-21 Name the service

3. Add a coach, called *ReviewClaims*, to the Human Service diagram, as shown in Figure 5-22 on page 135.

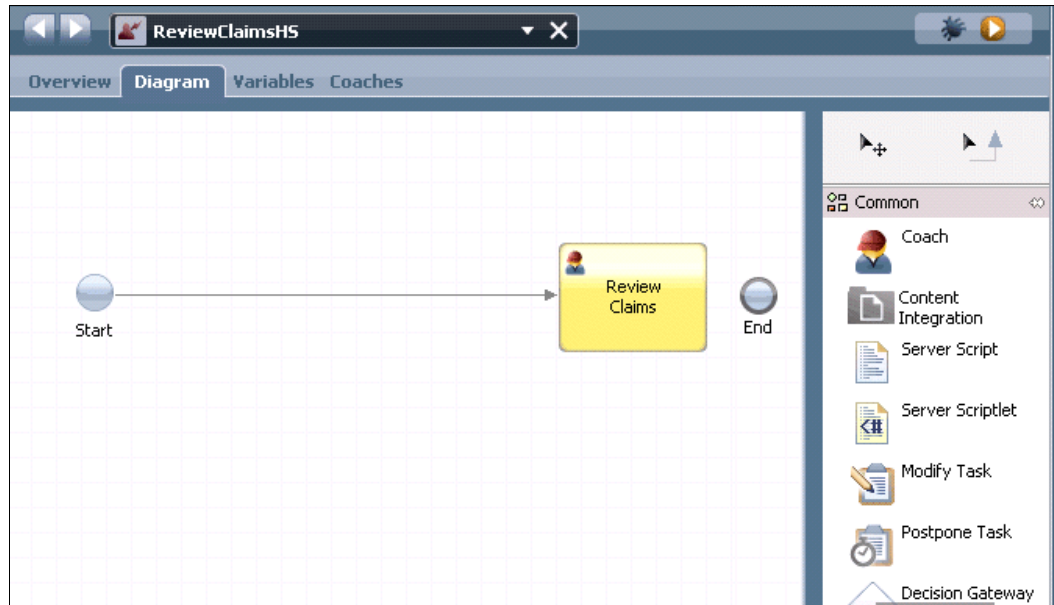


Figure 5-22 Add the coach to the human service diagram

4. Add the UI controls from the right side palette to the coach panel (Figure 5-23), and link the controls to input and output variables as needed.

This screenshot shows the 'Coaches' tab in the software interface. On the left, a list of input fields is displayed for the 'Review Claims' coach: 'SerialNumber 1', 'SecondsBetweenClaims 1', 'ProductType 1', 'ProblemType 1', 'Advice 1', and 'Status 1'. The 'Status 1' field is highlighted with a blue border and labeled 'Text: Text6'. On the right, a 'Coach View' palette is open, showing a list of UI controls: Button, Checkbox, Date Time Picker, Horizontal Section, Output Text, Select, Table, Tabs, Text (highlighted), and Vertical Section. Below the main workspace, a 'Properties' panel is visible, showing configuration options for the selected 'Text' control, including 'Label' (Status 1), 'Help', and 'Control Id' (Text6).

Figure 5-23 Add UI controls to the coach

After adding the UI controls, the ReviewClaims coach looks as shown in Figure 5-24.



Figure 5-24 ReviewClaims coach

5. Go to the Warranty Exception Process and link the Review Claims and Manager Review activities to the human service, as indicated in Figure 5-25 and Figure 5-26 on page 137.

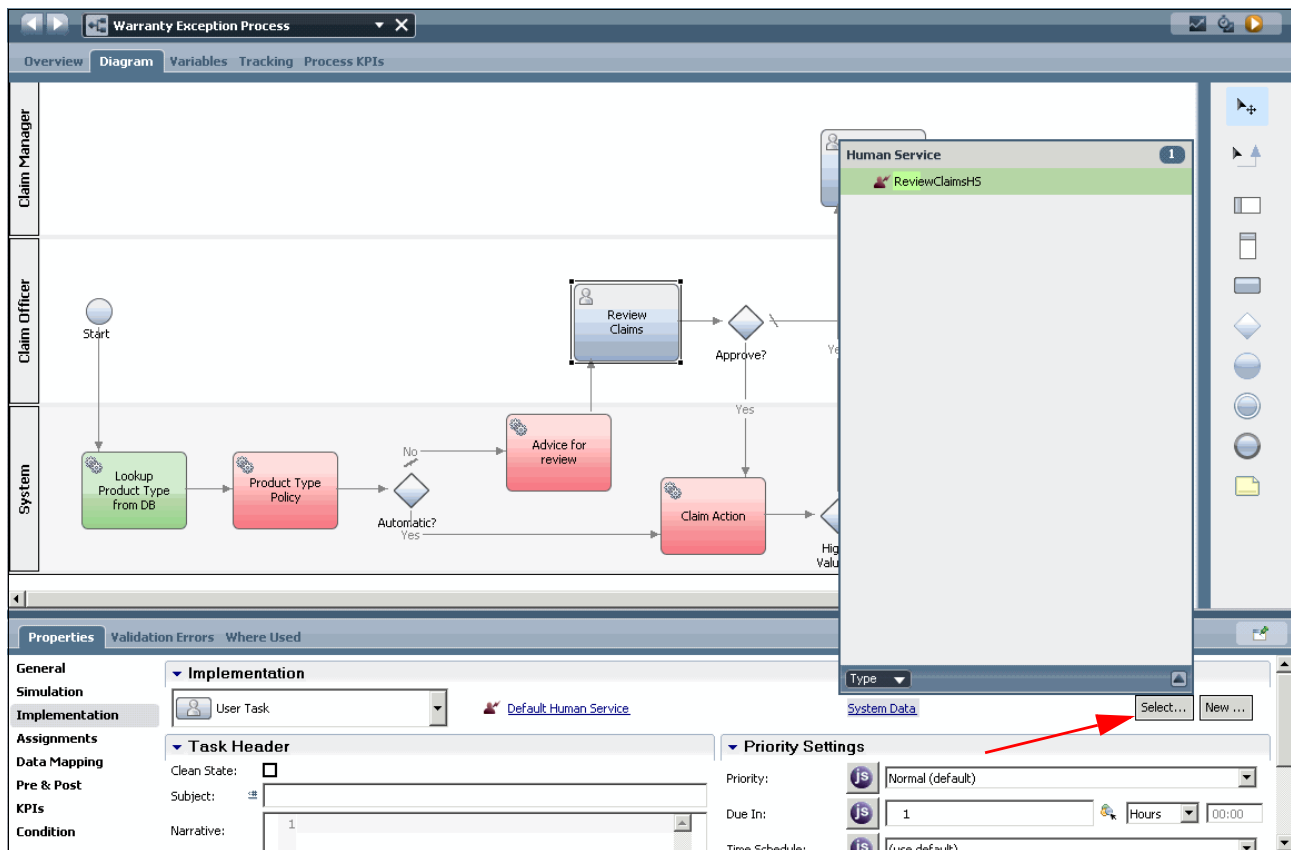


Figure 5-25 Link the human service to BPD activities

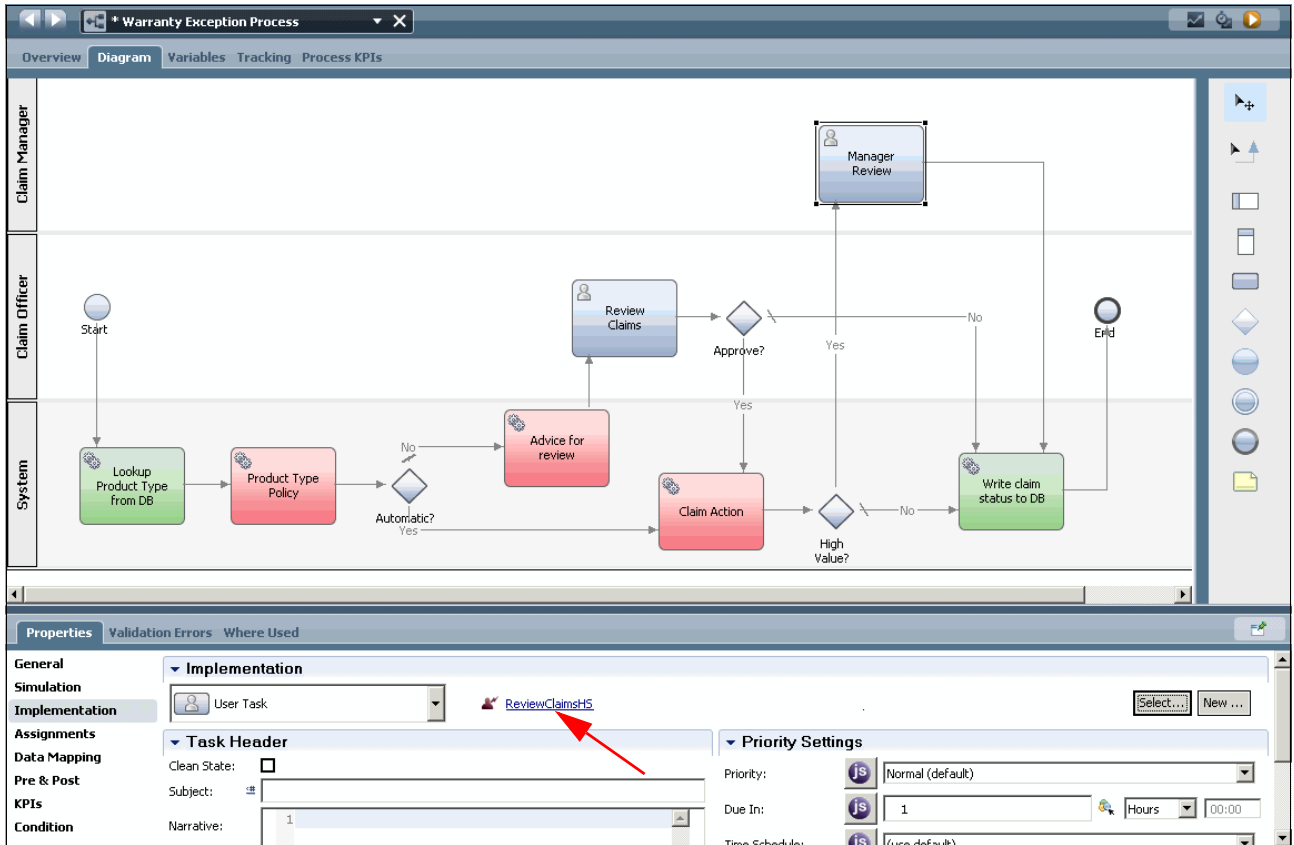


Figure 5-26 Link the human service to BPD activities

6. Modify the decision gateways (as shown in Figure 5-27) to use the output of the previous steps to determine the processing paths.

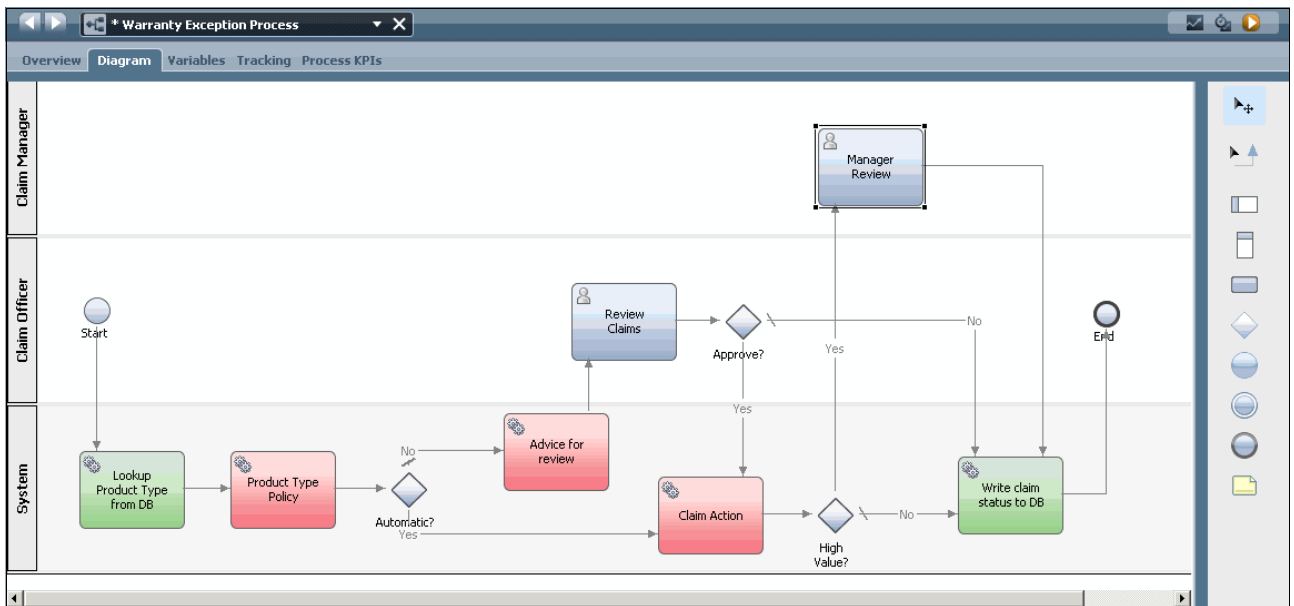


Figure 5-27 Modify the decision gateways

Now we have the event-related activities of the BPD linked to the business rules, and the decision gateways are using the information to split the process execution path. We have integrated the Business Process with the Business Rules in a loosely coupled way per the pattern “Business Process calling business rules” that is described in 4.2.1, “Decision management usage patterns” on page 80.

Note: For explicit guidance on setting up a user interface for your human task, we refer you to:

<http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wid.bpel.doc/topics/tnclient.html>

5.6 Enable a mediation to trigger the business process

The business process we have defined is triggered as follows:

1. A web user interface sends a serial number through a JMS queue.
2. The JMS message is received by a mediation flow, which converts the JMS message into a web service invocation.
3. The Warranty Exception Process is triggered when enough serial numbers have been received to raise an event.

Not all events require transformation and routing to successfully trigger a process, but our Warranty Claim example does. In this section we define a mediation that receives the JMS message that was produced as an action when the Warranty Claim Exception situation was detected by WebSphere Operational Decision Management. This mediation converts the JMS message to a web service invocation. We then document how to add a message-based trigger to our warranty exception process through the use of Undercover Agents and other services. The creation of the web user interface and JMS infrastructure is described in Appendix B, “Creating a messaging infrastructure and web user interface” on page 287.

5.6.1 Creating a new business object

Business objects define the business data that is used to describe the service and component contracts as well as the business data that the components manipulate. Basically, they represent the data that flows between services in an application. We need to define the data that will trigger the exception process.

To start, you need to define a new business object called `WarrantyEventType`.

1. Open IBM Process Designer and then open the process application you have been working on; if IBM Process Designer was already open, ensure you are in the Designer perspective.
2. Next to Data, click the plus button and select **Business Object**.
3. Name the Business Object `WarrantyEventType`.
4. Add the following fields to the `WarrantyEventType`:
 - `SerialNumber` with type `String`
 - `TimeBetweenEvents` with type `Integer`
 - `TimeOfLastEvent` with type `Date`
5. Save the business object.

5.6.2 Adding a message-based trigger to the business process

To create a message-based trigger, we will create the following general system services:

- ▶ Start Event
- ▶ Start Receiver

To begin:

1. Click the plus symbol next to the Implementation and select **General System Service**.
2. Name the new service Start Event, and click **Finish** to create the new service (Figure 5-28)

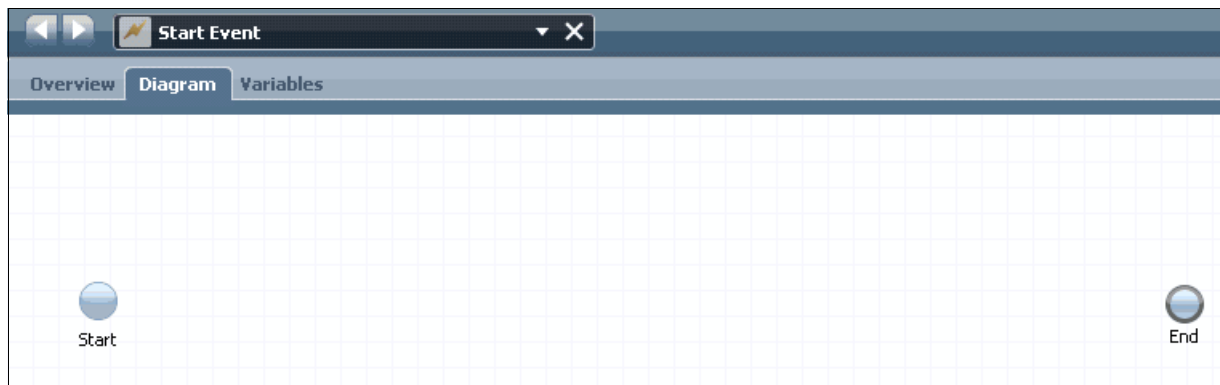


Figure 5-28 Start Event service diagram

3. Go to the Variables tab, click **Add Input**, name the variable WarrantyEvent and change its type to WarrantyEventType.
4. Next, click **Add Output**, name the variable WarrantyEvent and change its type to WarrantyEventType.
5. Go to the Diagram tab, and connect the start node directly to the end node using the arrow tool from the palette.
6. Save the activity.
7. Next to Implementation, click the plus symbol and select **General System Service**.
8. Name the new service Start Receiver (Figure 5-29).

Important: This is the second general system service that you need to create. It is important not to confuse the two services, so pay close attention to the name of the service that you are working in as you proceed.

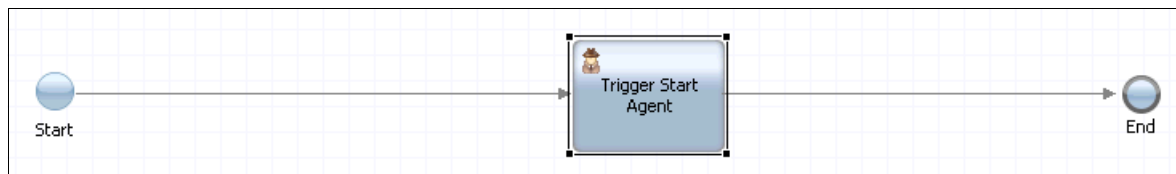


Figure 5-29 Start Receiver service diagram

9. Go to the Variables tab, click **Add Input**, name the variable WarrantyEvent and change its type to WarrantyEventType.

10. On the input parameter, select the **Has Default** option.
11. Click **Add Output**, name the variable `WarrantyEvent` and change its type to `WarrantyEventType`.
12. Go to the Diagram tab, drag an Undercover Agent (UCA) from the palette to the diagram and name the UCA activity `Trigger Start Agent`.
13. Go to the Implementation tab, click **New** to create a new UCA, as indicated in Figure 5-30, and name it `Start Agent`, as shown in Figure 5-31.

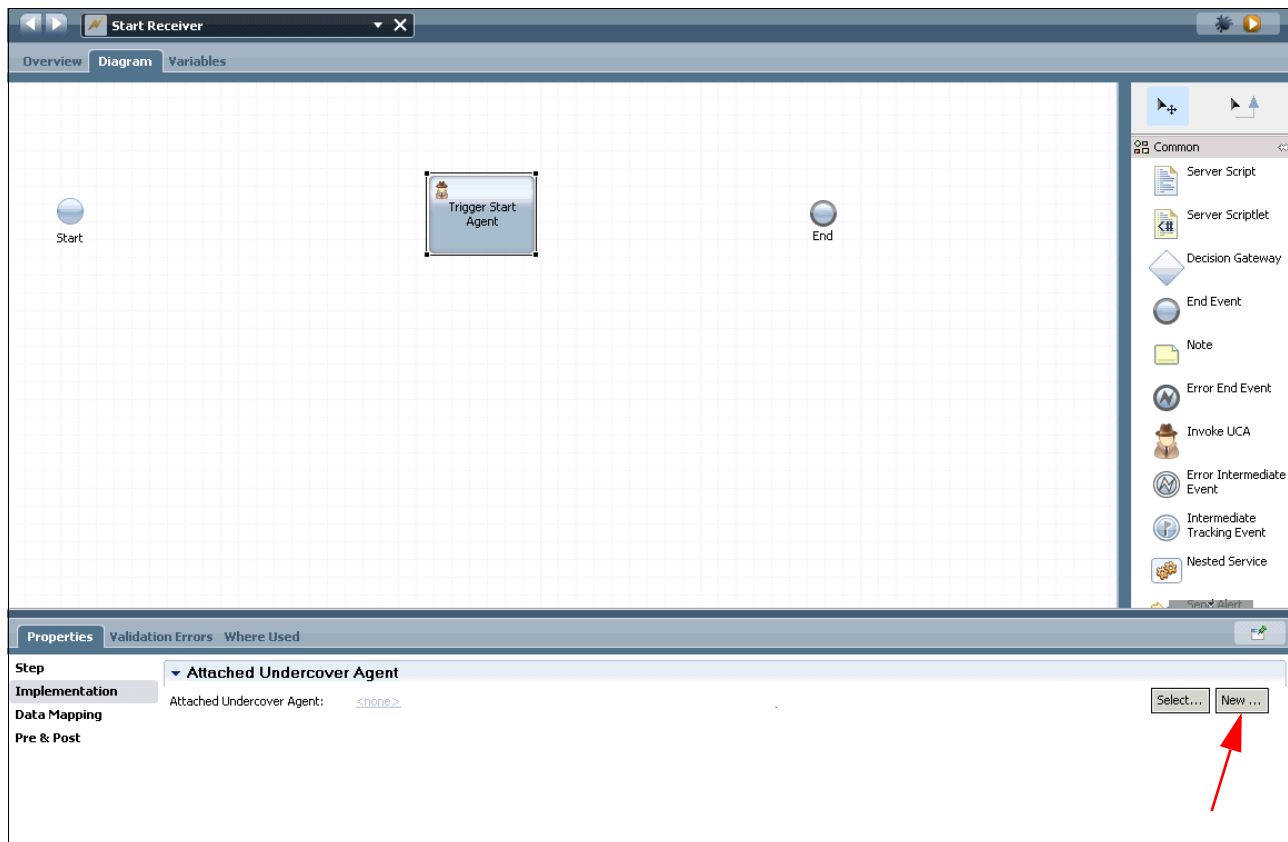


Figure 5-30 Create a new Under Cover Agent

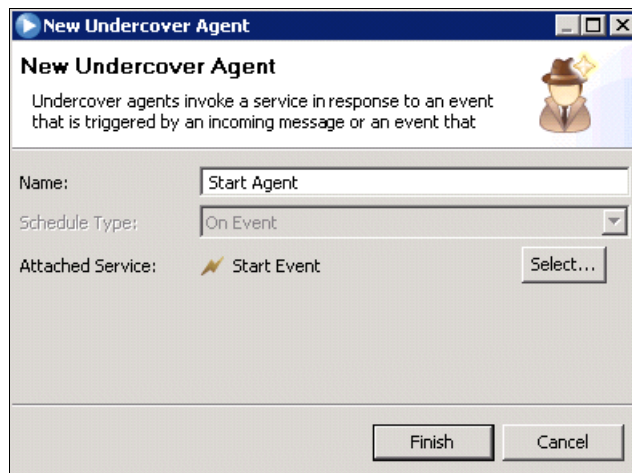


Figure 5-31 Create new Undercover Agent

Note: To learn about undercover agents and their use, we refer you to:

http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/using_undercover_agents.html

14. Click **Select** and choose the Start Event service.
15. Go to the Data Mapping tab and enter the value `tw.local.WarrantyEvent`.
16. On the diagram, connect the start node to the UCA Activity called Trigger Start Agent using the arrow tool on the palette.
17. Next, connect Trigger Start Agent to the end node in the same way.
18. Click the plus symbol next to Implementation, choose **Web Service** and name the new Web Service `StartWarrantyEventWS`.
19. In the Operations pane, click **Add**, and name the operation `StartWarrantyEventInBPM`.
20. Click **Select** and choose the **Start Receiver** service.

Now that the implementation is complete, you can create a new web service to expose the start event of Warranty Exception Process as a callable web service via SOAP.

1. First, attach the Start Receiver service to the newly created Web Service (as shown in Figure 5-32).

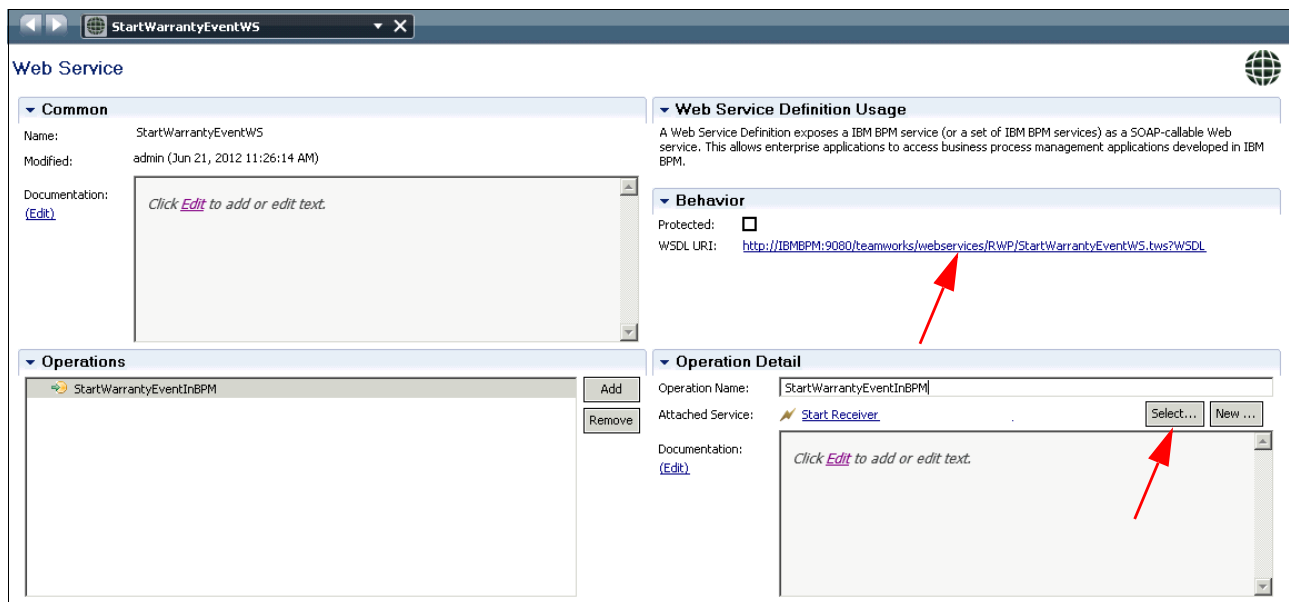


Figure 5-32 Attach the Start Receiver service to the new Web Service

2. Click the address <http://ibmbpm:9080/teamworks/webservices/RWP/StartWarrantyEventsWS.tws?WSDL> under Behavior to open the address in a new browser window, and display the WSDL.
3. Copy the address and save it in a text file (Figure 5-33).

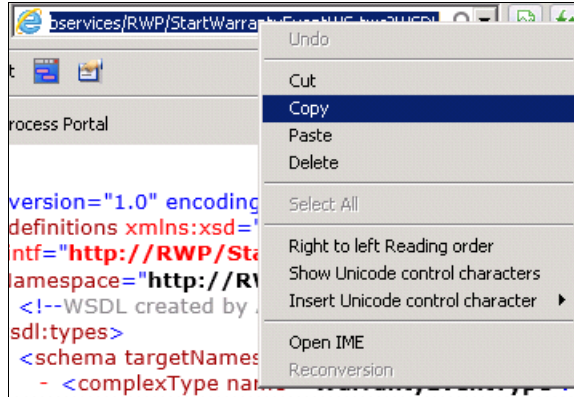


Figure 5-33 Open the WSDL in a browser and copy the web service address

4. Now, go the Warranty Exception Process and drag a start event into the diagram.
5. Name the event as Start Event, and place it at the start of the process, that is, before the Lookup Product Type from the DB activity.
6. Connect the Start Event to the next activity, that is, the Lookup Product Type from the DB activity (See Figure 5-34 on page 142).

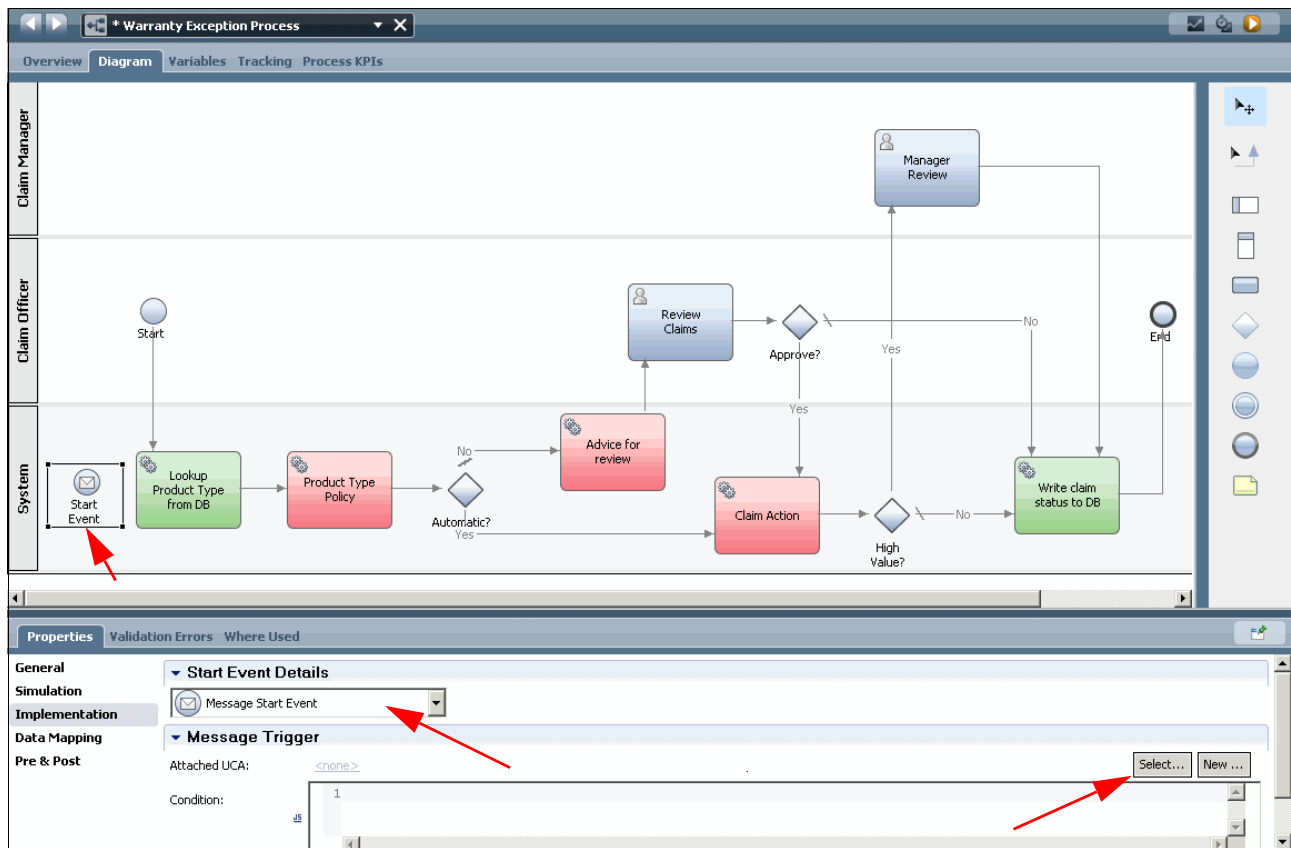


Figure 5-34 Connect the Start Event

7. Select the Start Event activity, and go to the Properties - Implementation pane.
8. Select **Message Start Event** from the Start Event Details drop-down menu, and attach the new UCA with this activity (See Figure 5-35).

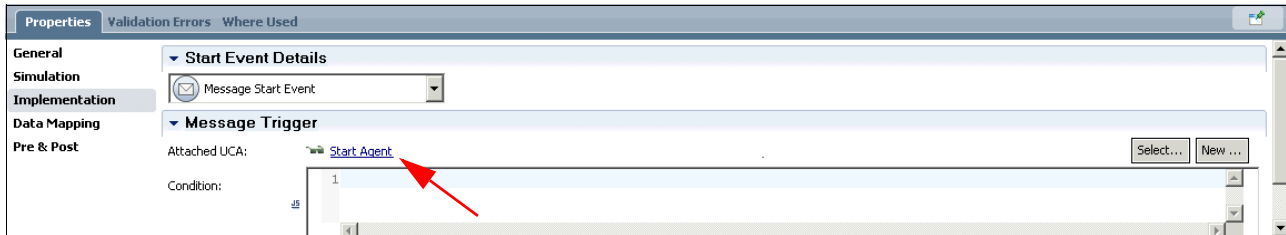


Figure 5-35 Attach UCA to the business process

5.6.3 Creating a mediation flow

A mediation is a way of mediating or intervening dynamically between services, and a mediation flow implements a mediation. We need to create a new mediation flow that exposes a JMS-based input interface and connects it with your newly created web service endpoint as the destination interface.

First open Integration Designer, and then open Process Center. It will look exactly as it does when you open it in Process Designer, so do not be concerned.

1. Open the Warranty Process application.
2. Then, right-click the process application and click **New** → **Mediation Module**.
3. Name the module `WarrantyMessageMediationModule`, and click **Next** (Figure 5-36).

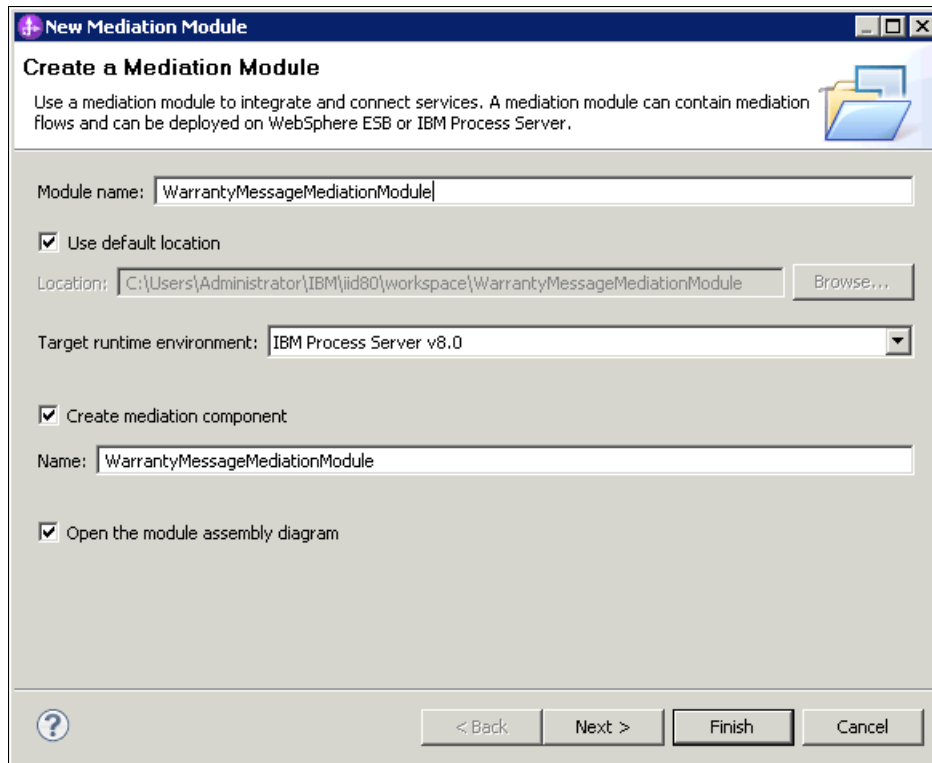


Figure 5-36 Create a mediation module

4. Ensure that **Eager Parsing** is selected and click **Next**.
5. Expand the **Business Integration** branch, select **WSDL and XSD** (Figure 5-37 on page 144) and click **Next**.

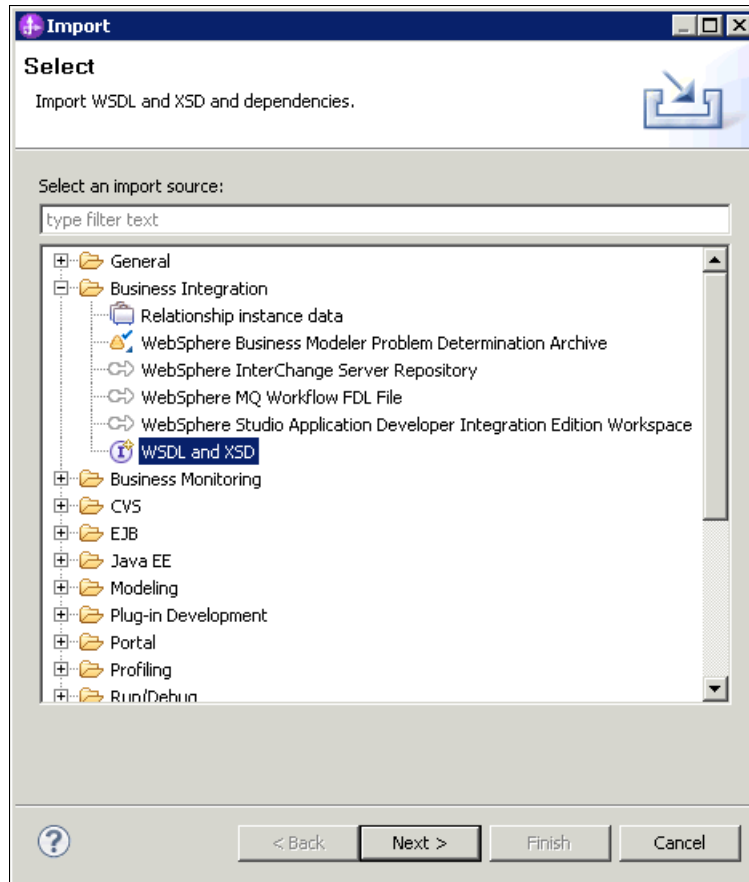


Figure 5-37 Select WSDL and XSD

6. Select **Remote WSDL or XSD file** (Figure 5-38) and click **Next**.

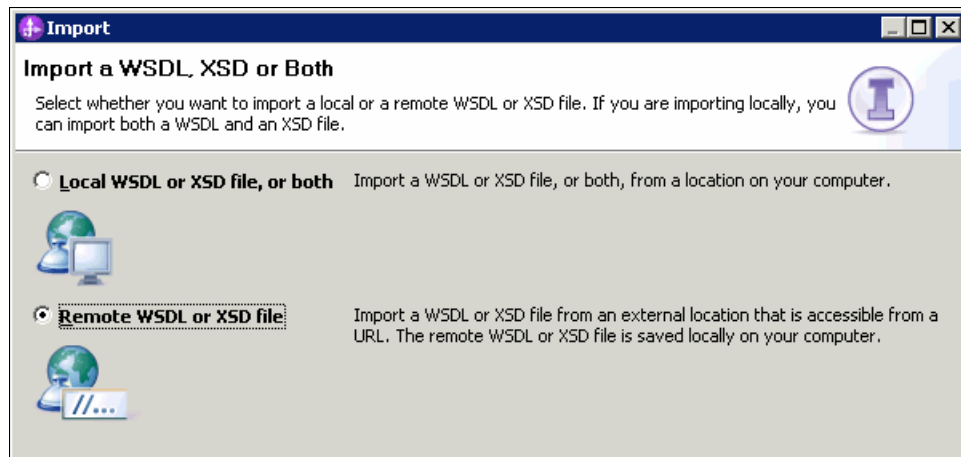


Figure 5-38 Select Remote WSDL or XSD file

7. Enter the address that you discovered in the Event Development in 4.5, “Overview of Event Rules” on page 92 and click **Next**.

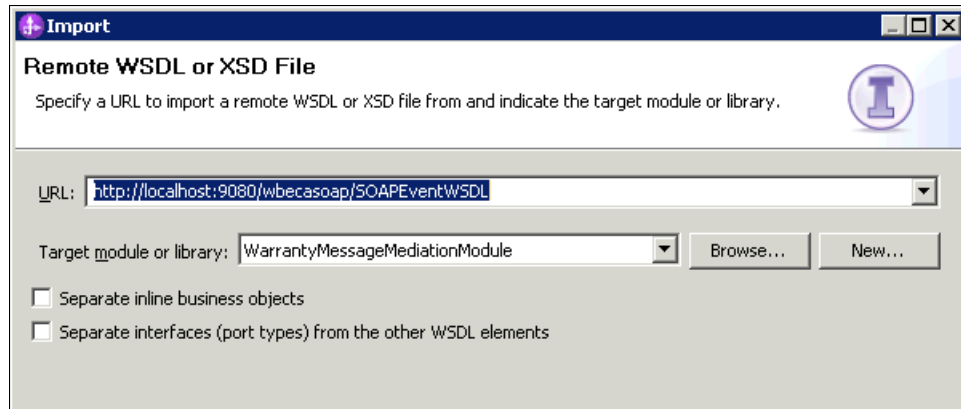


Figure 5-39 Import WSDL

8. Click **Finish**.
9. Add the association of the mediation module with the Warranty Processes application, as shown in Figure 5-40, and click **Next**.

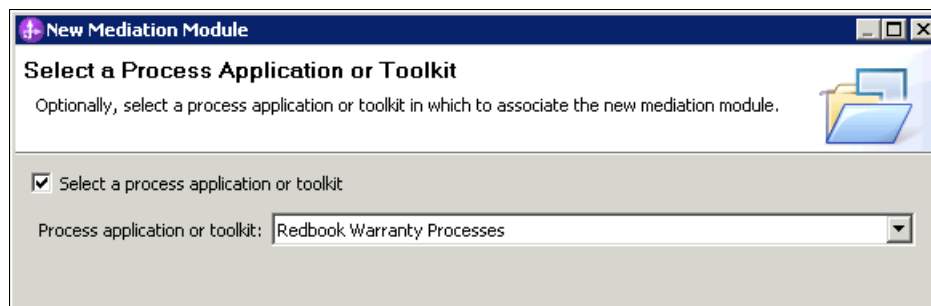


Figure 5-40 Select association with process application

10. Specify a transport protocol of **SOAP1.1/HTTP** (Figure 5-41 on page 146) and click **Finish**.

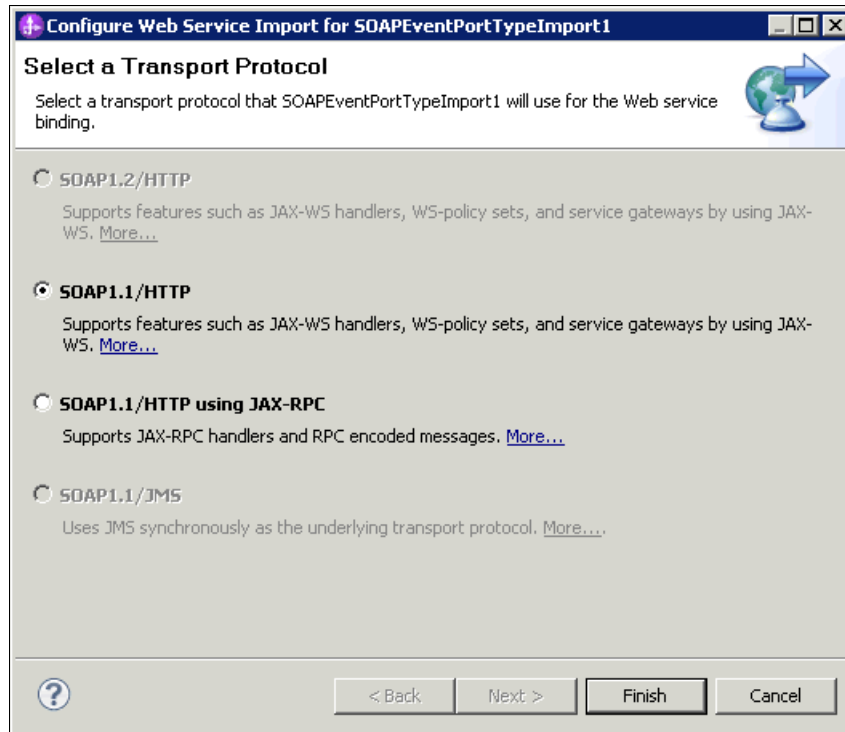


Figure 5-41 Select transport protocol

11. Select the SOAPEventPort, which is located under web service ports; drag it onto the assembly diagram, and note that an interface (SOAPEventPortType) and a data definition (EventReceived) are created (Figure 5-42).

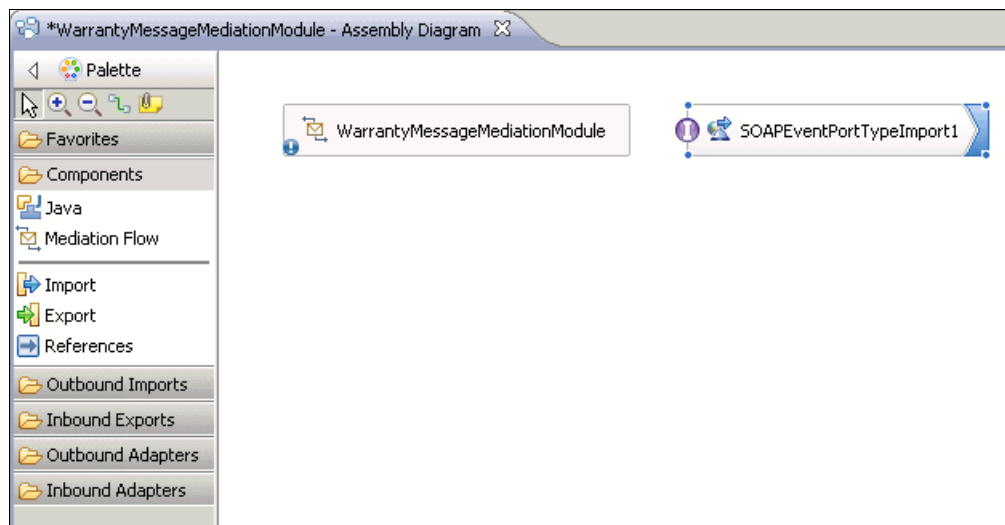


Figure 5-42 Warranty Message Mediation Module: Assembly diagram

12. Right-click the Mediation Module in the resource tree and select **New** → **Other**.
13. Under XML, select **XML Schema** (Figure 5-43). Click **Next**.

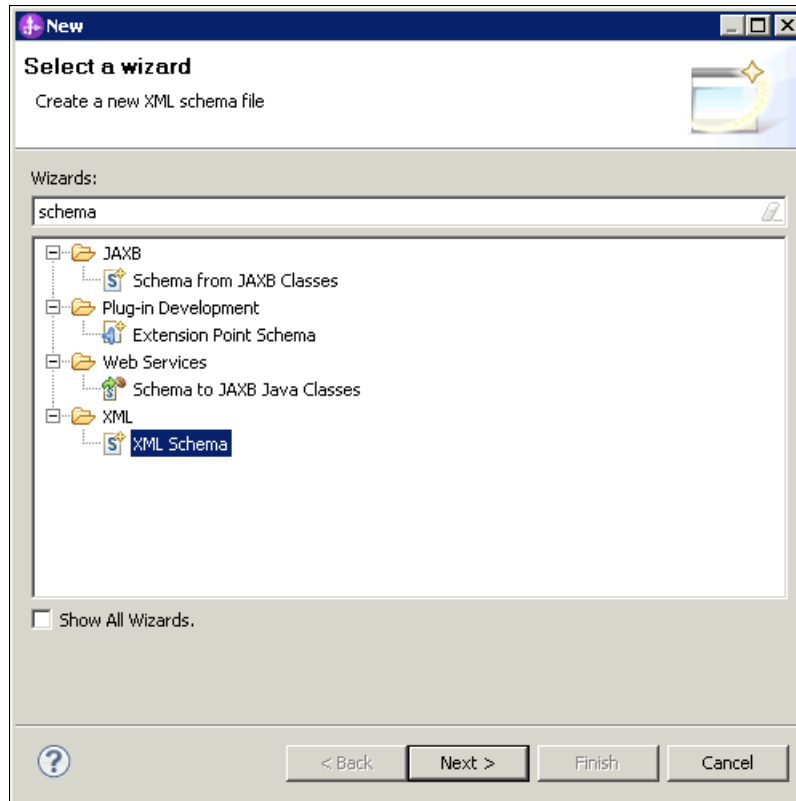


Figure 5-43 XML Schema

14. Name the file `input.xsd` and place it in the Mediation Module that you created.
15. When prompted that there was an error loading the business object (Figure 5-44), click **Yes** to open the XSD editor.

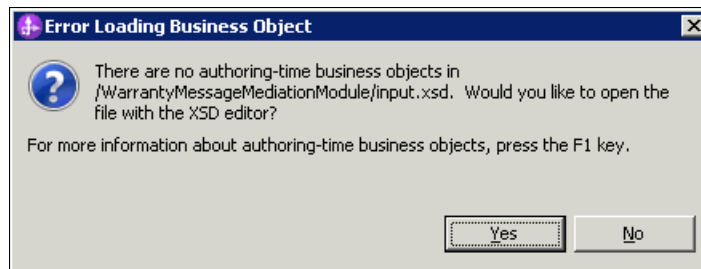


Figure 5-44 Error message: Error Loading Business Object

16. In the Source tab, paste the code shown in Example 5-1.

Example 5-1 `input.xsd`

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="CLUSTERID" type="xsd:string"/>
  <xsd:element name="TIMESTAMP" type="xsd:string"/>
  <xsd:element name="APPMESSAGE">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="CLUSTERID" />

```

```
        <xsd:element ref="TIMESTAMP" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

17. Save the XML file. A new entry is created in the Business Integration view called APPMESSAGE (Figure 5-45).

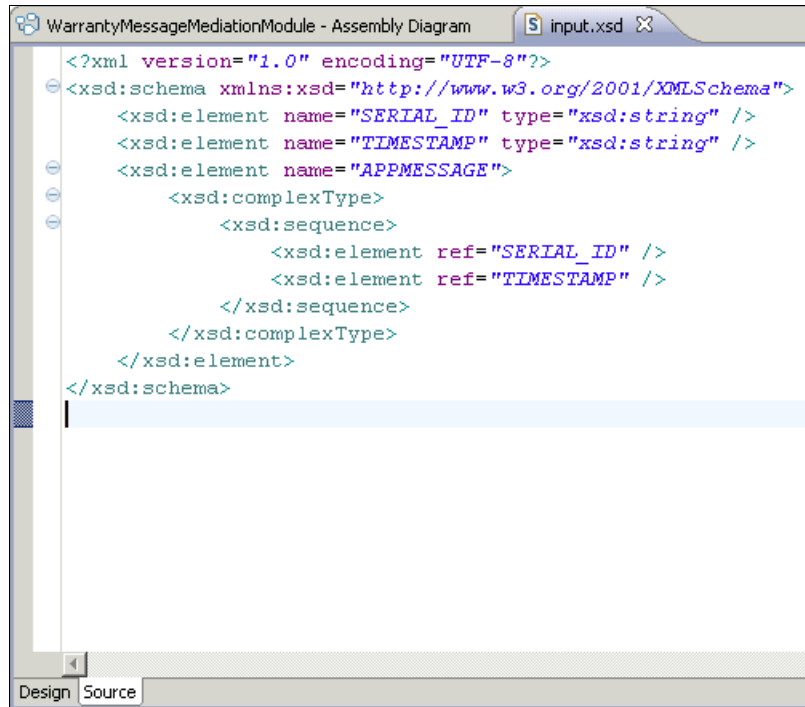


Figure 5-45 APPMESSAGE entry

18. Right-click **Interfaces**, choose **New** → **Interface**, and name the interface InputInterface (Figure 5-46 on page 149); click **Finish** when done.

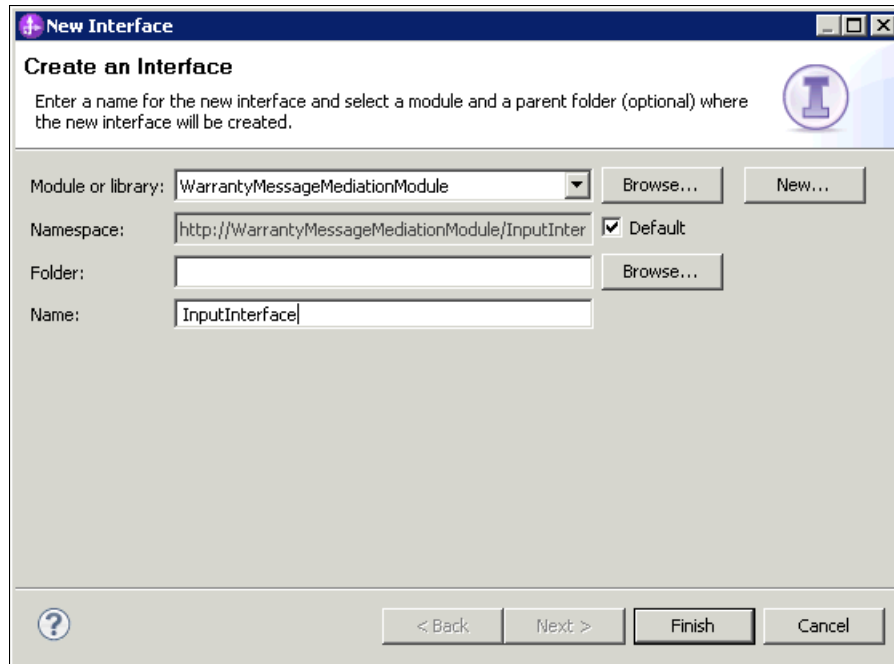


Figure 5-46 Create input interface

19. Select the operation and rename it to `inputWarrantyEvent`.
20. Change the type of the input to `APPMESSAGE`, and then save your interface (see Figure 5-47).

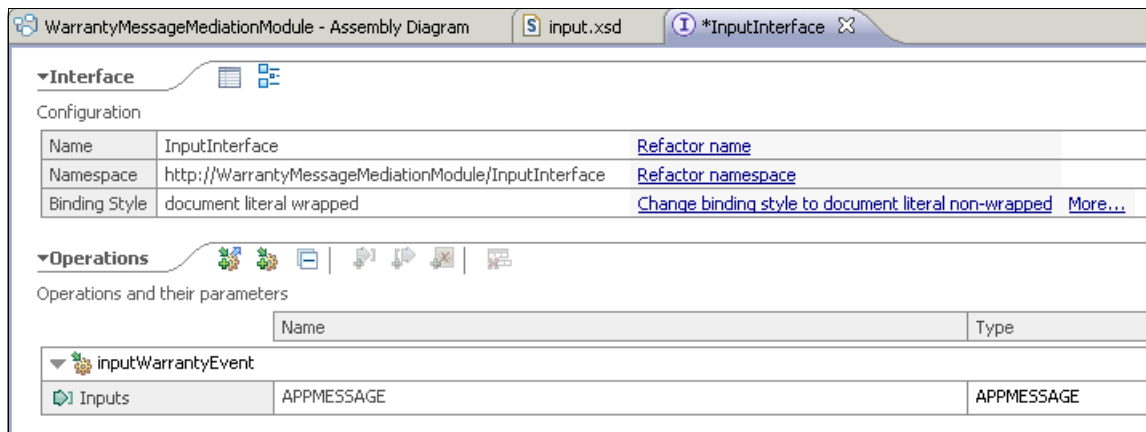


Figure 5-47 Create input interface

21. Next, open the Assembly diagram, and in the palette, expand the Inbound Exports.
22. Drag the JMS, *not* the Generic JMS, component to the left of the mediation module on the assembly diagram to open the Select an Interface dialog.
23. Click **Browse**, select **InputInterface** (Figure 5-48 on page 150), click **OK** followed by **Next**.

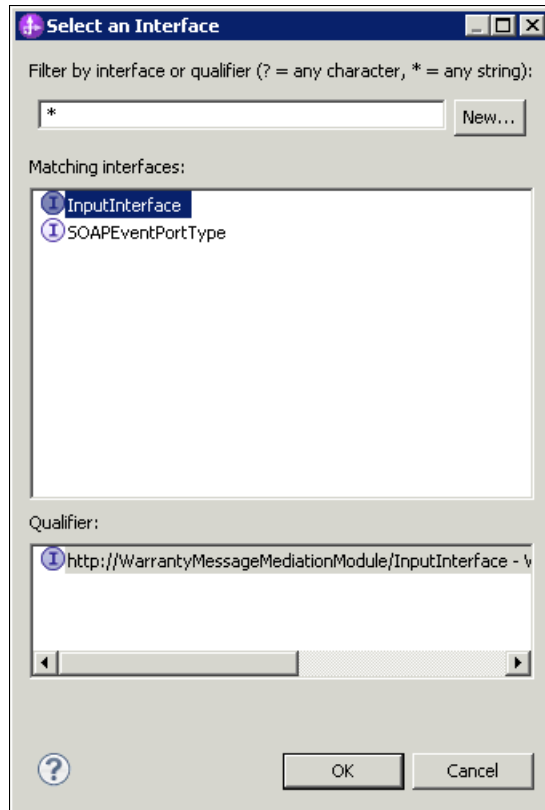


Figure 5-48 Select the Input Interface

24. For the JNDI name of the activation specification, select or type `.jms/actSpec`.
25. For the JNDI name of the receive destination, select or type `.jms/eventqueue` (Figure 5-49), and then click **Finish**.

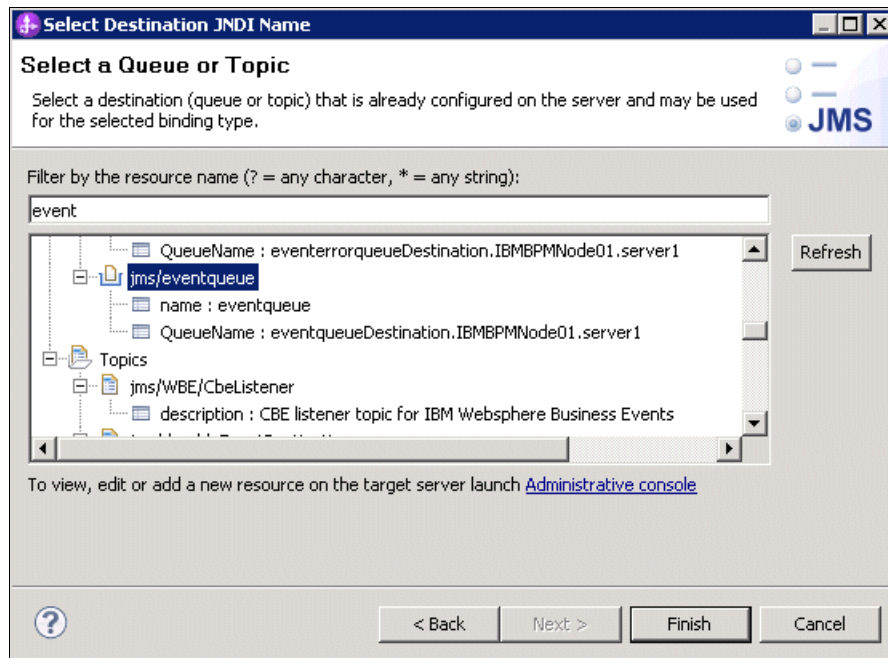


Figure 5-49 Select JMS queues

26. Wire the newly created export to the mediation by hovering over its tip and dragging the connector to the mediation.
27. When prompted whether you want the target service to be used in other modules (Figure 5-50), click **OK**.

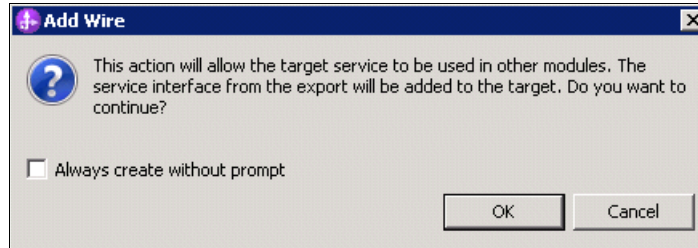


Figure 5-50 Add wire message box

28. Wire the mediation to SOAPEventPortTypeImport1.
29. When prompted whether you want the target service to be used in other modules (Figure 5-51), click **OK**.

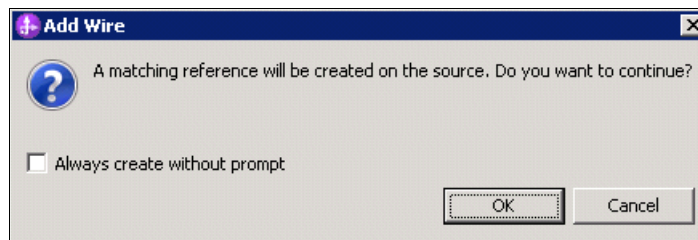


Figure 5-51 Add Wire

30. Save the mediation diagram.

With the mediation module created we now define the mediation flow:

1. Double-click the mediation module in the assemble diagram to implement it. When prompted, click **Yes** (Figure 5-52).

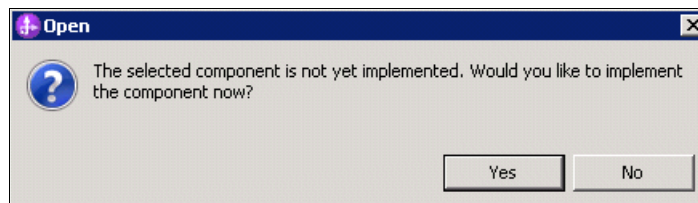


Figure 5-52 Generate Implementation

2. Choose the root folder, and click **OK** (Figure 5-53 on page 152).

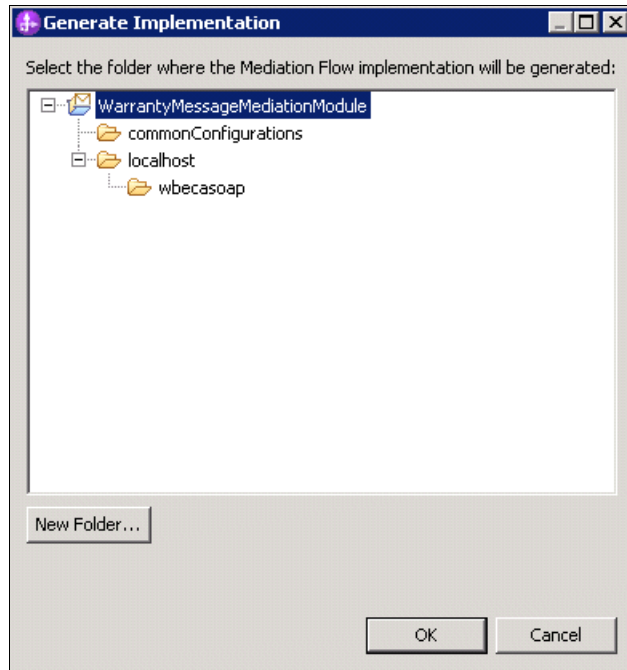


Figure 5-53 Generate Implementation

3. Click **inputWarrantyEvent**, and then select **Operation Map** (Figure 5-54).
4. Next, select **EventReceived** as the target operation, double-click the input map to implement it, and click **Next**.

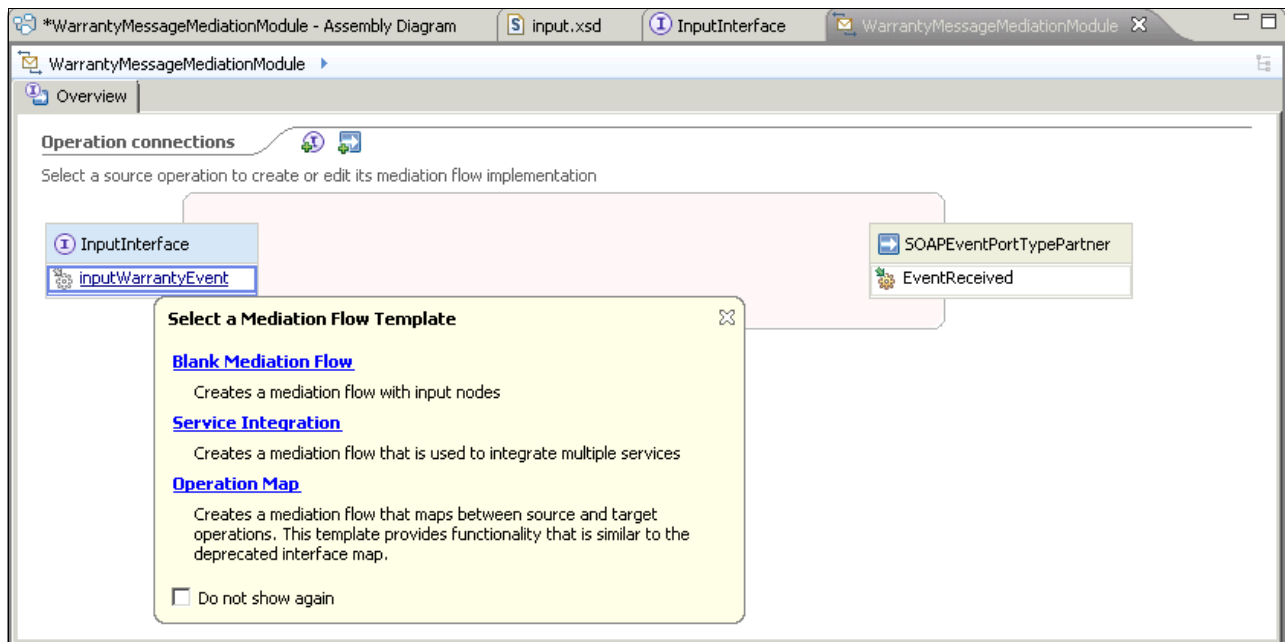


Figure 5-54 Generate Implementation

5. Click **Finish**.
6. Next, expand the APPMESSAGE and EventReceived objects on both sides of the input map so that you can see all of their members, and then drag SERIAL_ID to SerialNumber and TIMESTAMP to eventTime (Figure 5-55 on page 153).

7. Save all your changes.

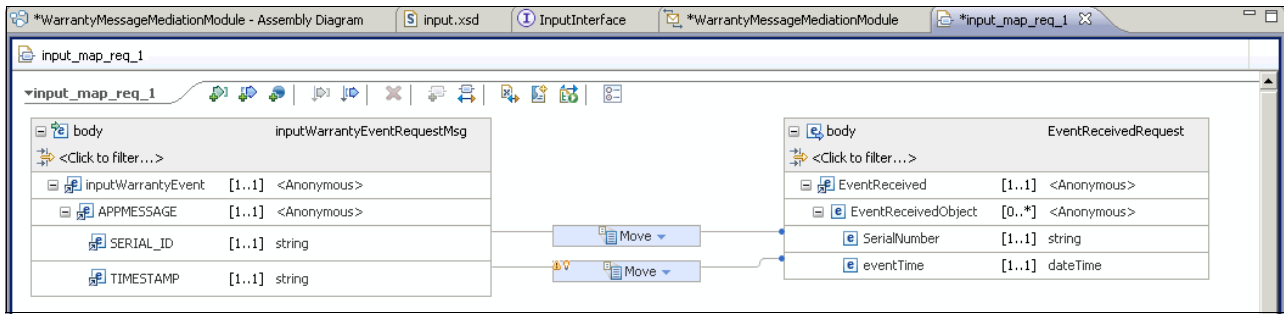


Figure 5-55 Create XML Map

8. In the Server Definition tab, right-click IBM Process Server and choose **Add and Remove**.
9. Select the Mediation Module that you created, and click **Add** to move it to the list of configured resources; this action deploys your mediation module.
10. Finally, associate the messaging infrastructure project with the Warranty Processes application that exists in the Process Center repository, as shown in Figure 5-56.

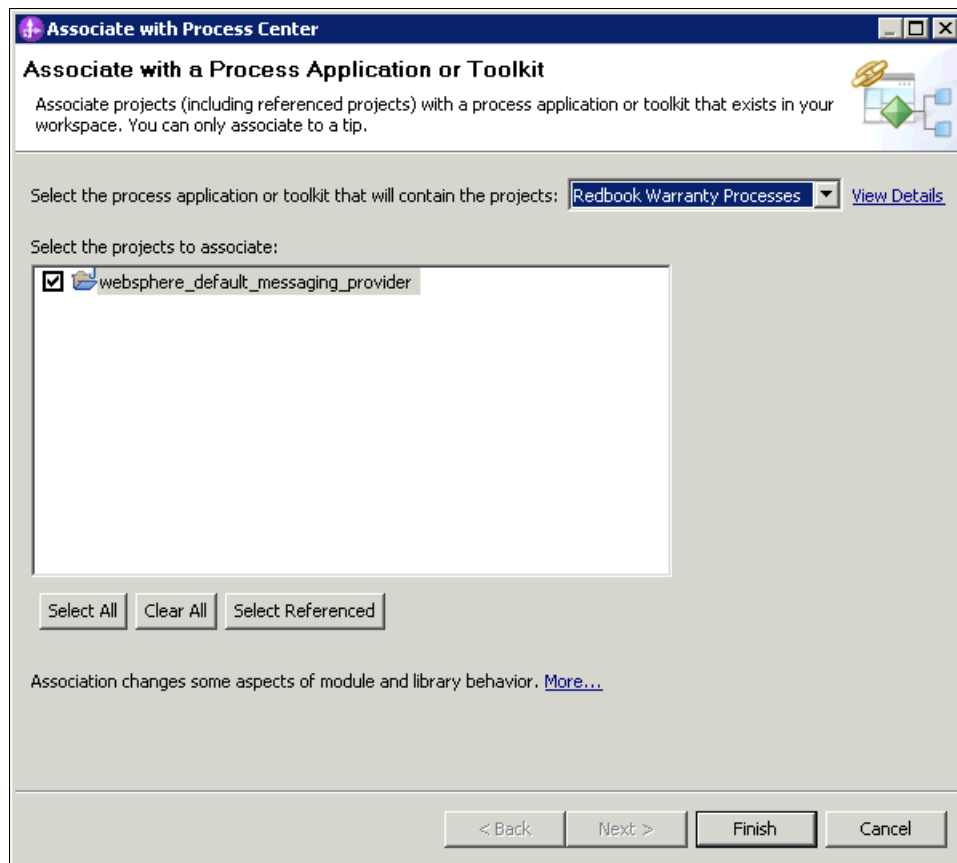


Figure 5-56 Associate messaging infrastructure project process application in Process Center

5.7 Add database integration

An integration service enables your processes to communicate with an external system to retrieve, update, or insert data. Since you want claims data (both approved and rejected claims) to be persisted into a database, you will create an integration service called `Persist` to perform that action.

Before starting, however, you need to have a database defined and a data source that references that database in WebSphere Application Server. The data table used is the `WARRANTY_TRANSACTION` table in the `WARRANTY` schema defined in Chapter 3, “Data modeling with InfoSphere Data Architect” on page 31. Alternatively, you can use the same steps with another database, replacing the appropriate parameters with your own database configuration details.

1. Open IBM Process Designer if it is not already open and ensure that you are in the Design view.
2. Click the plus symbol next to Implementation and select **Integration Service**.
3. Name the Integration Service `Persist`.
4. In the diagram, drag a Server Scriptlet, a Server Script, and a Nested Service onto the diagram from left to right.
5. From the right side palette, drag and add a Server Scriptlet, Server Script, and Nested Service activities onto the diagram from left to right and join them in sequence.
6. Name them, in the order that you dragged them (See Figure 5-57):
 - a. SQL statement definition
 - b. Set SQL parameters
 - c. SQL execute statement

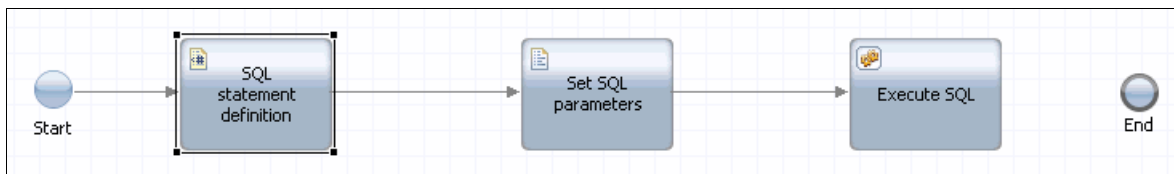


Figure 5-57 Add activities to the service diagram

7. Click the plus symbol next to Data, select **Business Object**, and call the new Business Object `TransactionType`.
8. Add the following parameters to this type by clicking **Add**:
 - WarrantyID Integer
 - CustomerID Integer
 - ProductSerialNumber Integer
 - WarrantyLevelID Integer
 - StatusID Integer
 - WarrantyDateTimeOccurrence Date
 - ClaimDescription String

Note: If you are using your own schema and table, above types must match those in the table you will insert into.

9. Open the Persist service that you created, select the Variables tab and add the following:
 - An input variable called Transaction with type TransactionType
 - A private variable called sqlStatement with type String
 - A private variable called sqlParameters with type SqlParameter
10. Select the **Is List** option for sqlParameters followed by the SQL statement definition.
11. Go to the Implementation tab and enter INSERT INTO WARRANTY.WARRANTY_TRANSACTION VALUE (?, ?, ?, ?, ?, ?, ?), or an SQL insert statement that matches your own schema; under binding, click **Select**. Select **tw.local.sqlStatement**.
12. Select **Set SQL parameters**.
13. Click the Implementation tab and copy in the code from Example 5-2 as shown in Figure 5-58 on page 156.

Example 5-2 SQL parameters

```

tw.local.sqlParameters = new tw.object.listOf.SqlParameter();
tw.local.sqlParameters[0] = new tw.object.SqlParameter();
tw.local.sqlParameters[0].value = tw.local.Transaction.WarrantyID;
tw.local.sqlParameters[0].type = "INTEGER";
tw.local.sqlParameters[1] = new tw.object.SqlParameter();
tw.local.sqlParameters[1].value = tw.local.Transaction.CustomerID;
tw.local.sqlParameters[1].type = "INTEGER";
tw.local.sqlParameters[2] = new tw.object.SqlParameter();
tw.local.sqlParameters[2].value = tw.local.Transaction.ProductSerialNumber;
tw.local.sqlParameters[2].type = "INTEGER";
tw.local.sqlParameters[3] = new tw.object.SqlParameter();
tw.local.sqlParameters[3].value = tw.local.Transaction.WarrantyLevelID;
tw.local.sqlParameters[3].type = "INTEGER";
tw.local.sqlParameters[4] = new tw.object.SqlParameter();
tw.local.sqlParameters[4].value = tw.local.Transaction.StatusID;
tw.local.sqlParameters[4].type = "INTEGER";
tw.local.sqlParameters[5] = new tw.object.SqlParameter();
tw.local.sqlParameters[5].value =
tw.local.Transaction.WarrantyDateTimeOccurrence;
tw.local.sqlParameters[5].type = "TIMESTAMP";
tw.local.sqlParameters[6] = new tw.object.SqlParameter();
tw.local.sqlParameters[6].value = tw.local.Transaction.ClaimDescription;
tw.local.sqlParameters[6].type = "VARCHAR";

```

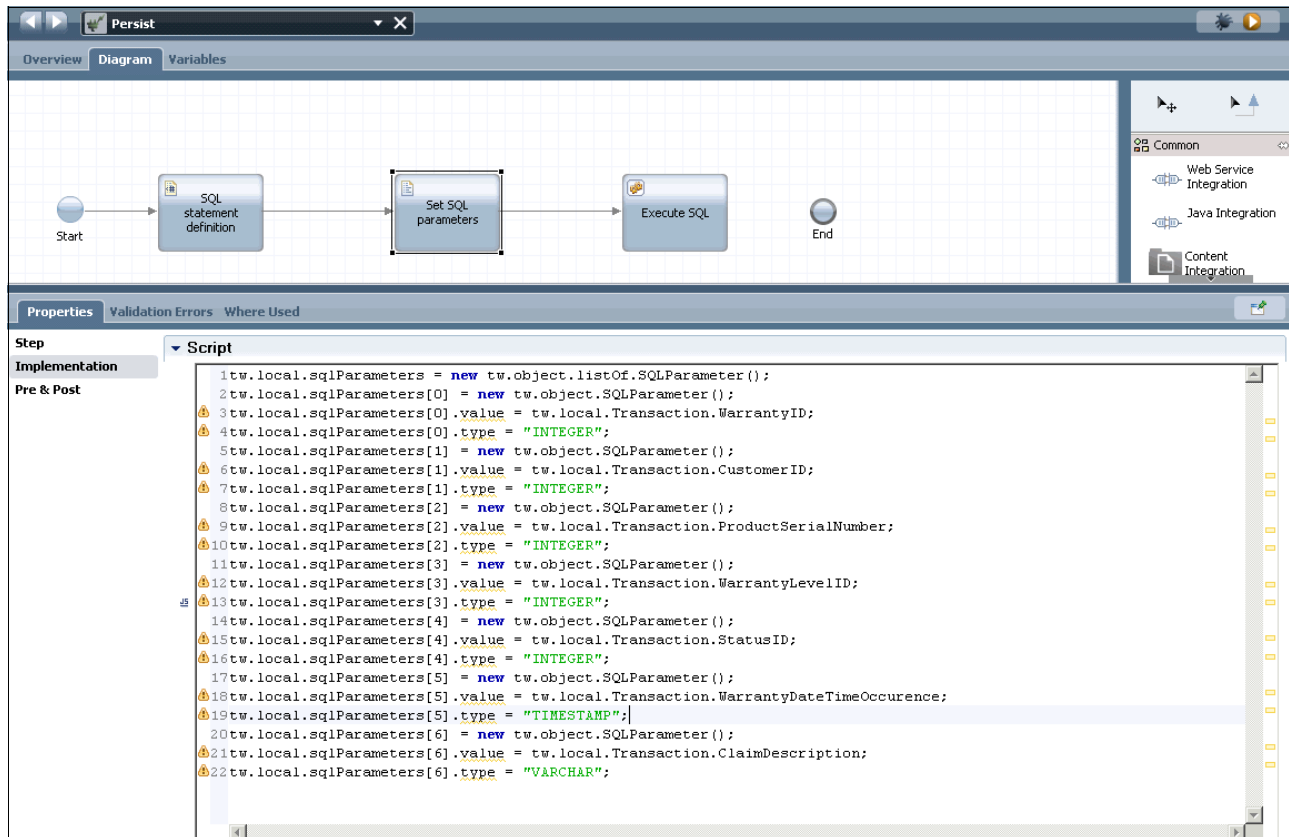


Figure 5-58 Set values into the SQL statement using Server Script

14. Select **ExecuteSQL**.
15. Go to the Implementation tab, and in attached service, click **Select**.
16. Choose **SQL Execute Statement**.
17. Go to the Data Mapping tab, next to SQL, enter `tw.local.sqlStatement`, and next to parameters, clear the default box and enter `tw.local.sqlParameters` (See Figure 5-59 on page 157).

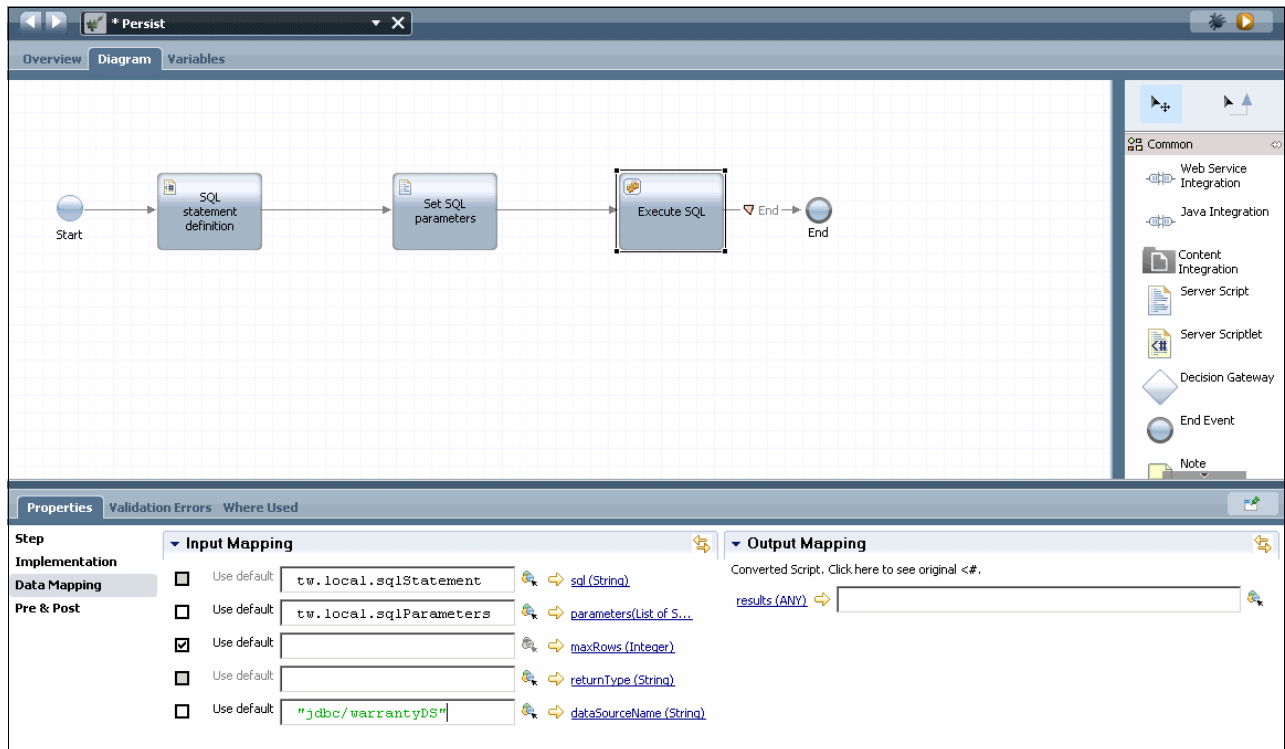


Figure 5-59 Add data mapping for the SQL Execute Statement

18. Next to `dataSourceName`, enter the data source that you have configured in WebSphere Application Server.
19. To create a simple test process, click the plus sign next to processes, select **Business Process Definition**, name the process **TestDatabase**, and click **Finish**.
20. In the diagram, drag the **Persist** activity from the integration list into the System swim lane.
21. Connect the start node to the **Persist** activity with the arrow tool, and connect the **Persist** activity to the end node with the arrow tool.
22. Go to the Variables tab, click **Add Input**, name the variable `transaction` and select the type `TransactionType`.
23. Click the **Has Default** option. Modify the default values to something nonconfidential, for example:
 - `WarrantyID: 101`
 - `CustomerID: 4099`
 - `ProductSerialNumber: 30005`
 - `WarrantyLevelID: 1`
 - `StatusID: 1`
 - `WarrantyDateTimeOccurrence: new TWDDate()`
 - `ClaimDescription "details"`
24. Go to the Data mapping tab, and for the input mapping, enter `tw.local.Transaction`.
25. Save the process.
26. Click the **Play** button, and if prompted, click **Yes** to switch views.
27. The data should now be in the database. You need to use any database tool that you have available to verify that the data has been inserted into the database.

5.8 Create an advanced integration service

Finally, we will create an advanced integration service, which is used to call a service implemented in Integration Designer from a BPD (via a system task) or another service (via a nested service). This service will confirm product types by serial numbers.

Using IBM Process Designer:

1. Create a new Advanced Integration Service called Product Lookup, then add an input field as SerialNumber (string) and an output field as ProductType (string) as shown in Figure 5-60.
2. Save the service.

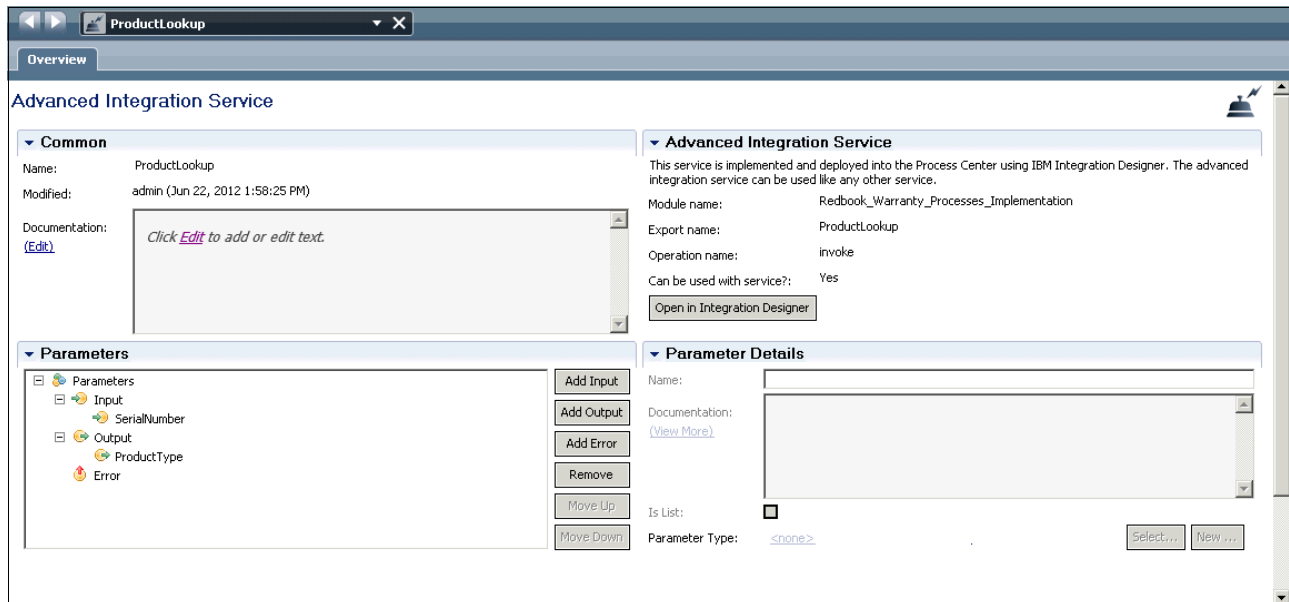


Figure 5-60 Create a new Advanced Integration Service

3. Open Integration Designer and go to Process Center, and then open the Warranty Processes process application (See Figure 5-61).

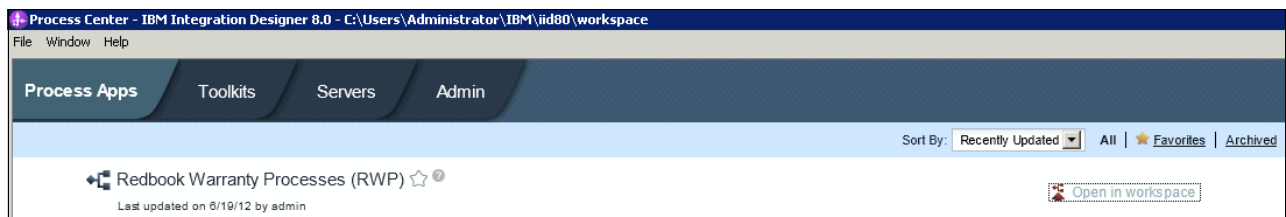


Figure 5-61 Open the Warranty Processes process application

4. Open the advanced integration, which shows that it is unimplemented; right-click it and choose **Implement** (See Figure 5-62 on page 159).

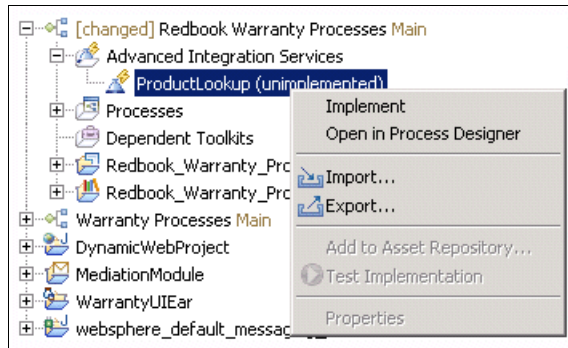


Figure 5-62 Implement the new Advanced Integration Service

5. Select a **Microflow** as the implementation type (See Figure 5-63) and add a snippet.

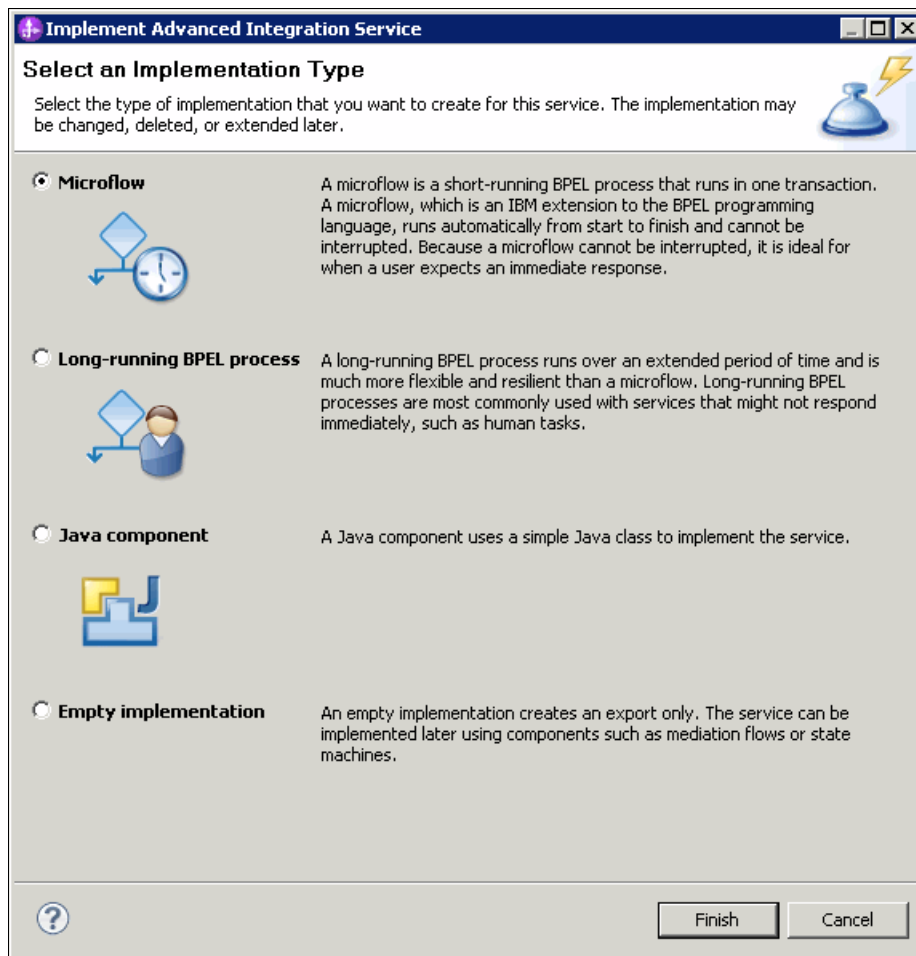


Figure 5-63 Choose Microflow as implementation type

6. In the properties for the snippet change to the **Java** source view and enter the code shown in Example 5-3.

Example 5-3 Snippet code

```
if (SerialNumber.startsWith("A")) {
```

```

        ProductType = "Apple";
    } else {
        ProductType = "Banana";
    }
}

```

7. Save the completed snippet as shown in Figure 5-64.

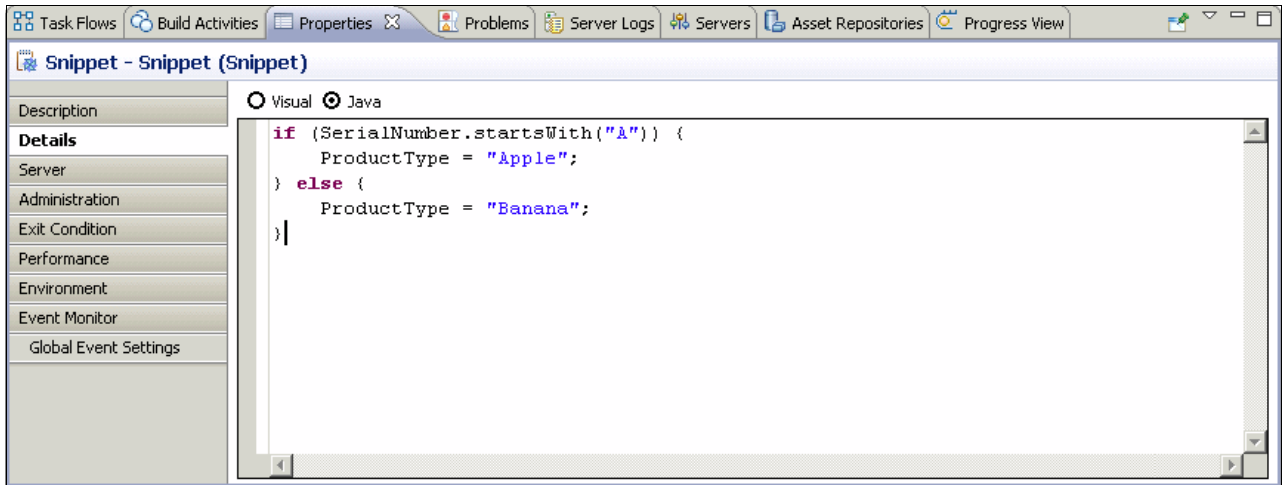


Figure 5-64 Add code to Java source view

8. Right-click and publish the process application.
9. Create a TestIntegration process (Figure 5-65), and add SerialNumber as input and ProductType as output (that is, the same as the Advanced Integration).

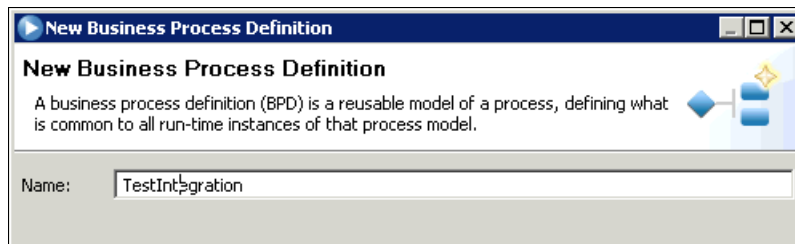


Figure 5-65 Create a TestIntegration process

10. Set the SerialNumber to any value as defaults (Figure 5-66).

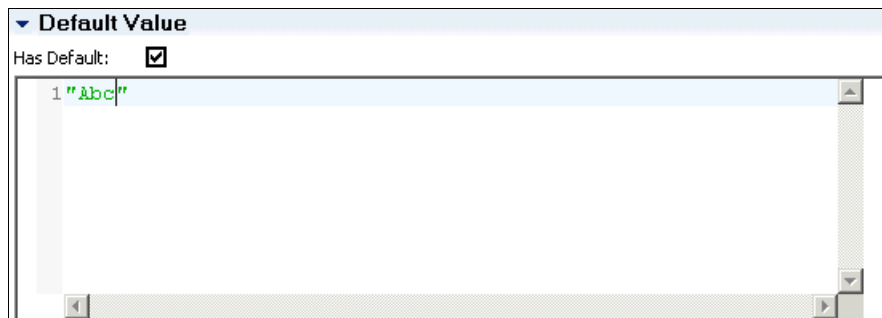


Figure 5-66 Set default for SerialNumber

11. Drag the Advanced Integration into the system swim lane (Figure 5-67).

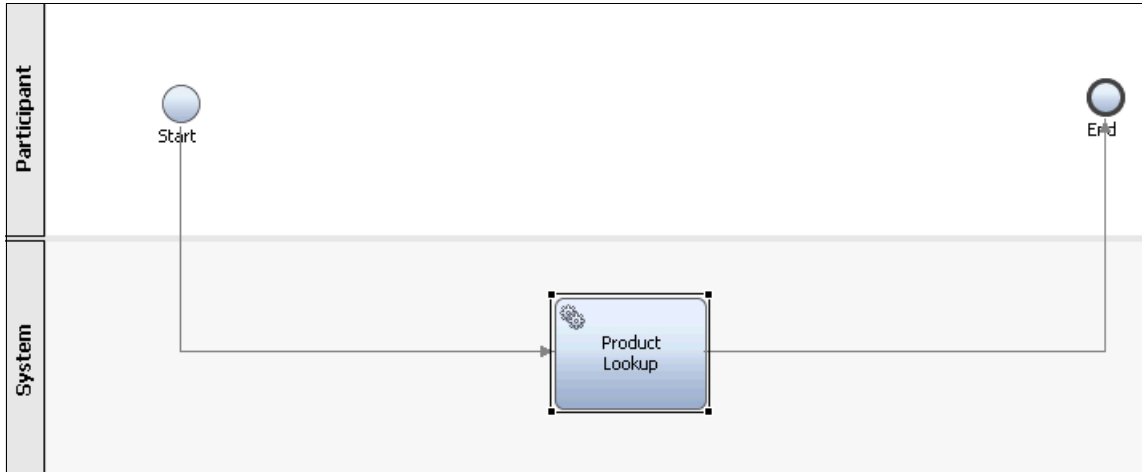


Figure 5-67 Drag the advanced integration service into the system lane

12. Hook the activity up to Start and End, and map the input and outputs to match (Figure 5-68).

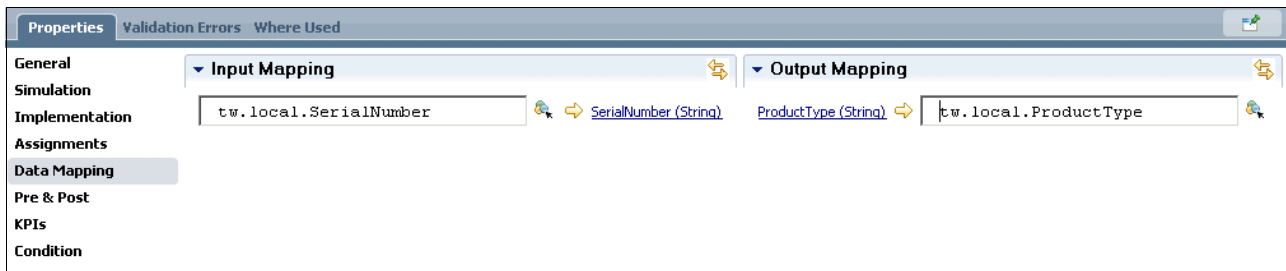


Figure 5-68 Map the inputs and outputs

13. Execute and examine the output to verify it (Figure 5-69 on page 162).

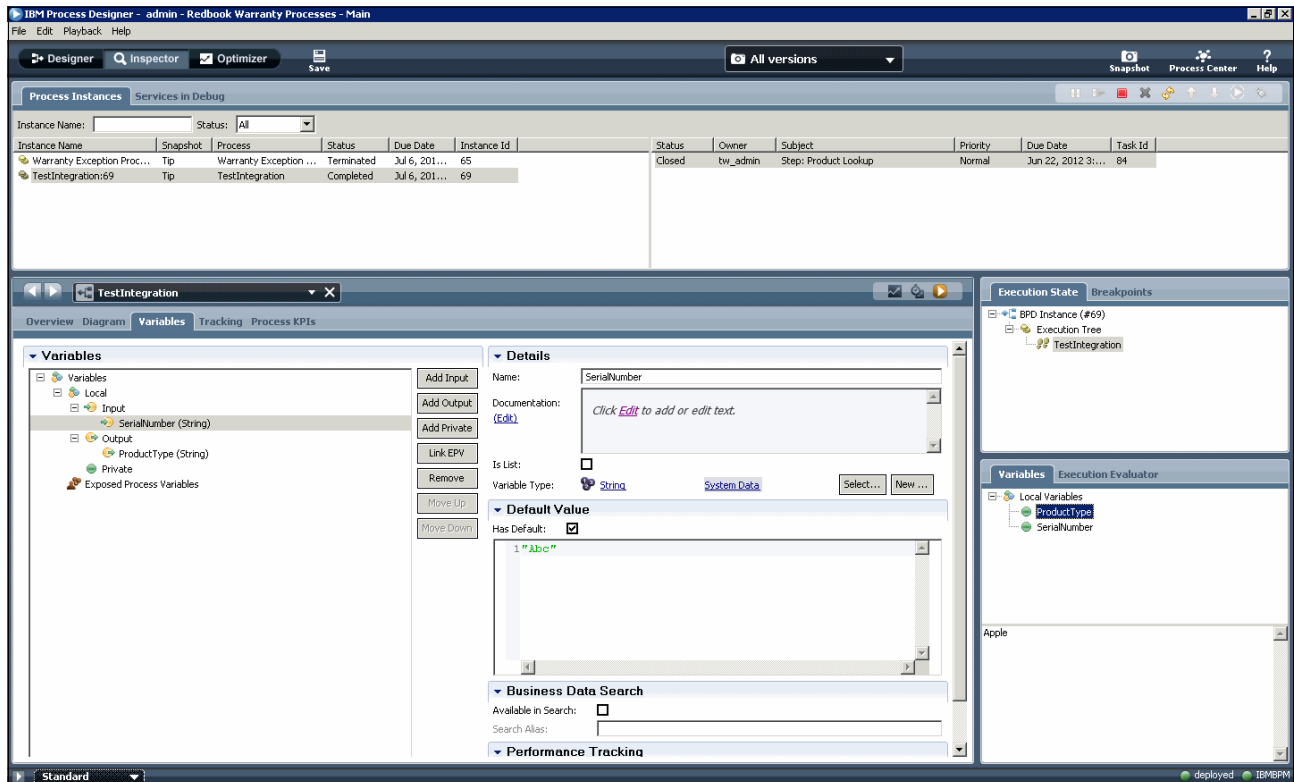


Figure 5-69 Execute the TestIntegration process and verify the output

For our warranty sample, we documented only one integration type in this chapter. Using IBM Business Process Manager, you can configure and manage outbound and inbound communication with external systems using BPM components and services to perform many tasks.

Outbound integrations enable business processes authored in Process Designer to interact with other systems, such as a web service, a content management system, or an external database. Depending on the system that you are integrating with, you can implement the integration service using an Integration Service implementation or an IBM Case Manager Integration Service implementation.

Note: To learn more about creating outbound integrations, we refer you to:

http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/creating_outbound_integrations.html

For inbound integrations that involve an external system or application calling into IBM Business Process Manager to kick off a service, you need to build several BPM components and corresponding services.

Note: To learn more about creating inbound integrations, we refer you to:

http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/creating_inbound_integrations.html

5.9 Understanding Task Types and Service Types

The following tables list the various task types and service types supported by IBM Business Process Manager. Tasks are components that allow people or services to interact, while services are how you enable these interactions.

Table 5-4 lists the types of tasks that are available in IBM Business Process Manager.

Table 5-4 Types of tasks available in IBM Business Process Manager

Task Type	Task Description	Associated Service Types
User Task	<p>User tasks should be completed by process participants and so are associated with Human services by default. When you drag a Human service from the library to a BPD diagram, IBM Process Designer automatically creates an activity with a User task with the Human service selected.</p> <p>When you drag an activity from the palette to a participant lane in a BPD diagram, IBM Process Designer automatically creates an activity with a User task with the Default Human service selected. For cases where you want a user to start the service but no additional user involvement is required, you can also choose a User task type and associate a service with it, such as an Integration or Advanced Integration service. In this way, IBM Process Designer automatically creates the user implementation that you need when you drag process components onto a diagram. You can also choose User Task and an associated service for an activity's implementation in the properties pane.</p>	Human service, Integration Service, Advanced Integration Service
System Task	<p>System tasks should be completed by an automated system or service and are automatically performed without requiring user initiation regardless of the type of lane in which they reside in a BPD diagram. When you drag an Ajax service, General System service, Integration service, or Advanced Integration service from the library to a BPD diagram, IBM Process Designer automatically creates an activity with a System task type, regardless of whether the service is dragged to a system lane or a participant lane.</p> <p>Dragging an activity from the palette to a system lane in a BPD diagram automatically creates an activity with a System task with the Default System service selected. System tasks that you place in a non-system lane are also performed by the system. In this way, IBM Process Designer automatically creates the System implementation that you need when you drag process components onto a diagram. You can also choose System Task and an associated service for an activity's implementation in the properties pane.</p>	Ajax service, General System Service, Integration Service, or Advanced Integration Service
Decision Task	<p>Decision tasks are useful when you want a decision or condition in a business rule to determine which process implementation is invoked. When you drag a Decision service from the library to a BPD diagram, IBM Process Designer automatically creates an activity with a Decision task. You can also choose Decision Task and an associated Decision service for an activity's implementation in the properties pane.</p> <p>Note: Decision tasks in IBM Business Process Manager are equivalent to BPMN 2.0 Business Rule Tasks.</p>	Decision service

Similar to tasks, the type of service that you choose to create depends upon the requirements of the activity. For example, if an activity requires integration with an external system, such as

a database, you can create an Integration service. If an activity requires call center personnel to enter data about customer requests, you can create a Human service with a coach.

Table 5-5 describes the types of services that are available in IBM Business Process Manager.

Table 5-5 Types of services available in IBM Business Process Manager

Service Type	Description
Decision Service	<p>Use a Decision service when you want a condition to determine the implementation invoked. For example, when a certain condition evaluates to true, IBM Business Process Manager implements the JavaScript expression that you provide. Decision services cannot include Java or Web Service integrations directly. You can call a Decision service from any other type of service and a Decision service can call other nested services.</p> <p>To learn more about building a Decision service, we refer you to: http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/topics/crule_ruleserv_ovw.html</p>
Human Service	<p>Use a Human service when you want to create an interactive service. A Human service is the only type of service that can contain coaches and postpones. Human services generate tasks in IBM Process Portal.</p> <p>Note: A Human service is the only type of service that can call other nested Human services.</p> <p>Note: The Postpone Task saves the current execution context of a Human service and closes any associated open coaches. When the task is opened from IBM Process Portal, execution starts at the exit port of the Modify Task control.</p> <p>To learn more about building a Human service, we refer you to: http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/building_human_service.html</p>
Ajax Service	<p>Use an Ajax service when you want to include a control in a coach to implement dynamic data selection such as automatically populating drop-down lists and automatically completing edit boxes. An Ajax service can pull data dynamically from a connected data source, such as a database. You cannot call an Ajax service from other types of services, but an Ajax service can call other nested services.</p> <p>To learn more about building an Ajax service, we refer you to: http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/building_ajax_service.html#building_ajax_service</p>
Integration Service	<p>Use an Integration service when you want to integrate with an external system. An Integration service is the only type of service that can contain a Java or Web Service integration. You can call an Integration service from any other type of service and an Integration service can call other nested services.</p> <p>To learn more about building an Integration service, we refer you to: http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/building_integration_service.html</p>
Advanced Integration Service	<p>Use an Advanced Integration service when you want to integrate with a service created in Integration Designer.</p> <p>To learn more about building an Advanced Integration service, we refer you to: http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/tadvintegrationsrvbld.html#tadvintegrationsrvbld</p>

Service Type	Description
IBM Case Manager Integration Service	<p>Use an IBM Case Manager Integration service when you want to integrate with an IBM Case Manager server.</p> <p>To learn more about building IBM Case Manager Integration service, we refer you to: http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/cicmintegrationsrv.html</p>
General System Service	<p>Use a General System service when you need to coordinate other nested services or you need to manipulate variable data. For example, if you need to implement data transformations or generate HTML for a coach, you can use a General System service. General System services cannot include Java or Web Service integrations directly. You can call a General System service from any other type of service and a General System service can call other nested services.</p> <p>To learn more about building a General System service, we refer you to: http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/topic/com.ibm.wbpm.wle.editor.doc/modeling/topic/building_gen_system_service.html#building_gen_system_service</p>

Understanding how each type supports BPM will enable you to become more effective at modeling processes with IBM Business Process Manager.



Reporting with IBM Cognos Business Intelligence

In this chapter we describe how to use IBM Cognos Business Intelligence to elicit insight from our warranty sample.

Business intelligence is defined as the ability to gather, analyze and report data in a way that enables business leaders to make better decisions. Companies that implement business intelligence effectively are best positioned to succeed in today's fast-paced business world as well as to outperform their competition.

Creating a successful IBM Cognos Business Intelligence application requires careful planning and assessment, because you need to understand the questions vexing your business users while confirming the limitations of your data or processes.

The sections in this chapter define how specific roles can work with Cognos Business Intelligence interfaces to gather, analyze and report data effectively.

6.1 Overview of IBM Cognos Framework Manager

IBM Cognos Framework Manager is a metadata modeling tool that drives query generation for IBM Cognos software. This client tool for Windows workstations allows a metadata modeler to create a business representation and “friendly” view of the metadata for report authors and business users, who generally do not understand the underlying structure and names of objects, such as databases, schemas or catalogs, tables, and columns that are used in a source data system. The tool helps hide the complex structure that is used to access and represent the data by creating package definitions that can be used across capabilities within IBM Cognos Business Intelligence.

A *package* can contain the full contents of the model or a logical subset tailored for a specific user community or line of business. The result is an environment where tables and column names are presented in a meaningful and logical manner in the various authoring interfaces for the target audience.

6.1.1 Data access strategies

IBM Cognos Business Intelligence can connect various sources of data, ranging from transactional databases, data warehouses, data marts, online analytical processing (OLAP) systems, enterprise resource planning (ERP) systems, to web services so that business users can extract meaningful results and answer pressing questions. This capability is powerful, but you also need to have a viable data access strategy to make these tools useful. When evaluating data access strategies in context of the reporting requirements and the desired user experience, consider the following questions:

- ▶ How fresh does the data need to be?
- ▶ Is the source appropriate for reporting?
- ▶ Are there interactivity requirements from the users?
- ▶ What are expected data volumes, especially considering data growth over time?
- ▶ Is the source data movable to another system or structure?
- ▶ Is this application going to be mission critical?
- ▶ Is the source system able to handle the load this application may generate?

Answering these types of questions can help you to determine the appropriate source requirements.

For example, if your reporting requirement calls for minute-by-minute data and data is being updated all the time, you will likely want a transactional source. However, if the reporting requirement is less frequent, such as daily or weekly, you can offload your data to a data warehouse or data mart. Regardless of the data source chosen, IBM Cognos Business Intelligence is designed to optimize the query experience against all supported data sources. IBM Cognos Query layer generates the queries in a manner that is suited to the capabilities of the given source system.

Our warranty sample uses IBM DB2 Universal Database™ V9.7 as a transactional database system and IBM Cognos Business Intelligence V10.1.1 Fix Pack 1 with Dynamic Query mode. These choices match the requirements for transactional level reporting and for the use of Business Insight, which allows for interaction with the results.

6.1.2 Metadata model

A model is a collection of metadata that includes physical information and business information for one or more data sources. The construction of a metadata model supports the

creation and publishing of reporting packages. These reporting packages enable users to be able to quickly locate, access, and report on data.

When creating your metadata model, it is best to understand the reporting requirements of your business. The greater number of requirements you can cover in a model, the less maintenance your model will require over time, which can save you time and money over the long run. Before you make a model available to your user community, be sure that it is complete per the user requirements. If a model must be published early, before it is final, you will want to leverage the IBM Cognos Business Intelligence model versioning capabilities. These capabilities will help you to keep track of all the versions of your model so you can reconcile them when your model is final.

6.2 Creating data source connections

The warranty sample makes use of the new dynamic query mode to enable query optimizations to our IBM DB2 Universal Database via a single connection definition. This connection is defined using the IBM Cognos Administration interface, without the need for any thick client installation.

To create the data source connections you need to:

- ▶ Configure IBM DB2 connectivity
- ▶ Create the connection to the warranty claims database

6.2.1 Configure IBM DB2 connectivity

To enable the reporting engine to connect to IBM DB2, or another relational DB, you only need to copy the Java Database Connectivity (JDBC) driver files to the IBM Cognos installation directory.

To configure IBM Cognos Business Intelligence with the IBM DB2 JDBC drivers:

1. Locate and copy the installed `db2jcc4.jar` and `db2jcc_license_cu.jar` files from the `../sqllib/java` directory on your IBM DB2 server.
2. Next, locate the `../v5dataserver` and `webapps ../p2pd/web-inf/lib` directories on the IBM Cognos Business Intelligence servers with the Report Server component installed, and paste in both the `db2jcc4.jar` and `db2jcc_license_cu.jar` files.
3. Restart each IBM Cognos Business Intelligence server where the files were copied.

Note: To learn more about setting up the dynamic query mode, go here:

http://publib.boulder.ibm.com/infocenter/cbi/v10r1m0/topic/com.ibm.swg.ba.cognos.s.dyn_query.10.1.0.1.doc/c_dyn_query_setup.html#dyn_query_setup

6.2.2 Create the connection to the warranty claims database

The data in the warranty claims database allows you to create reports detailing customer warranty claims. These reports can help business users assess the need for making adjustments to their processes, products or businesses.

Before you begin: IBM Cognos Administration is part of the Cognos web interface and can be accessed from IBM Cognos Connection using the Launch menu.

To create the connection to the warranty claims database via the Cognos Administration interface:

1. Click the **Configuration** tab followed by the **New Data Source** icon (Figure 6-1) to start the New Data Source wizard.

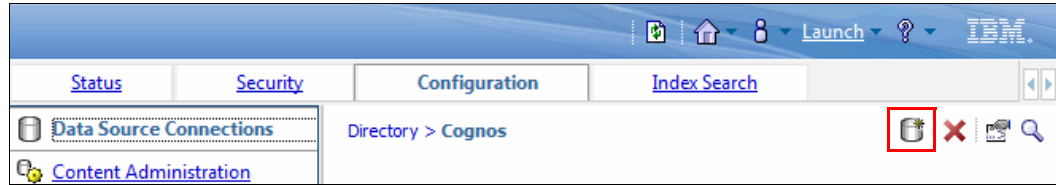


Figure 6-1 New data source

2. Enter the following information in the New Data Source window (Figure 6-2) and click **Next** when done:

Name	Warranty claims database
Description	Warranty claims database records all incoming requests
Screen tip	Warranty program database

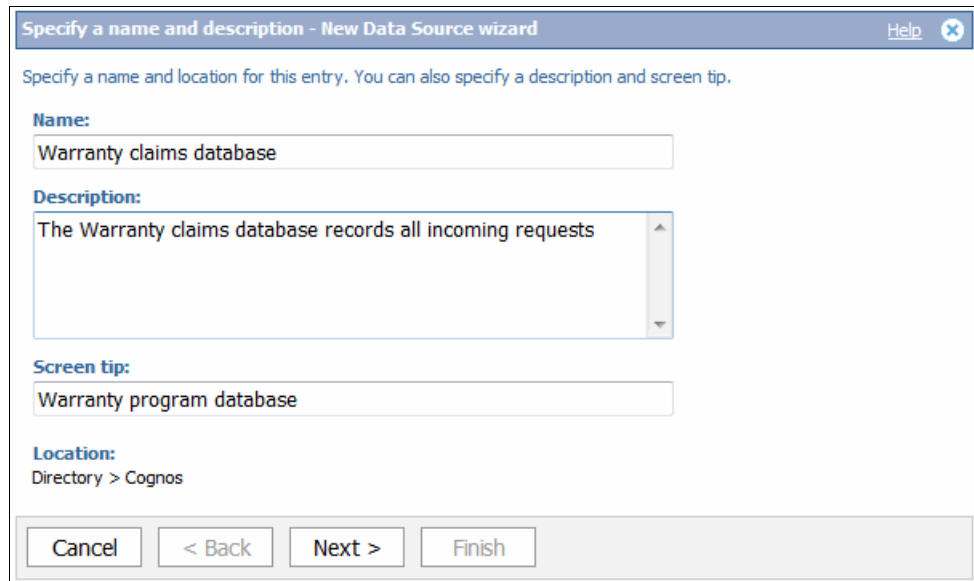


Figure 6-2 New Data Source wizard: Specify a name and description

3. In the **Specify the connection** dialog that opens, select **Type** → **IBM DB2**, and check the **Configure JDBC connection** checkbox as shown in Figure 6-3 on page 171.

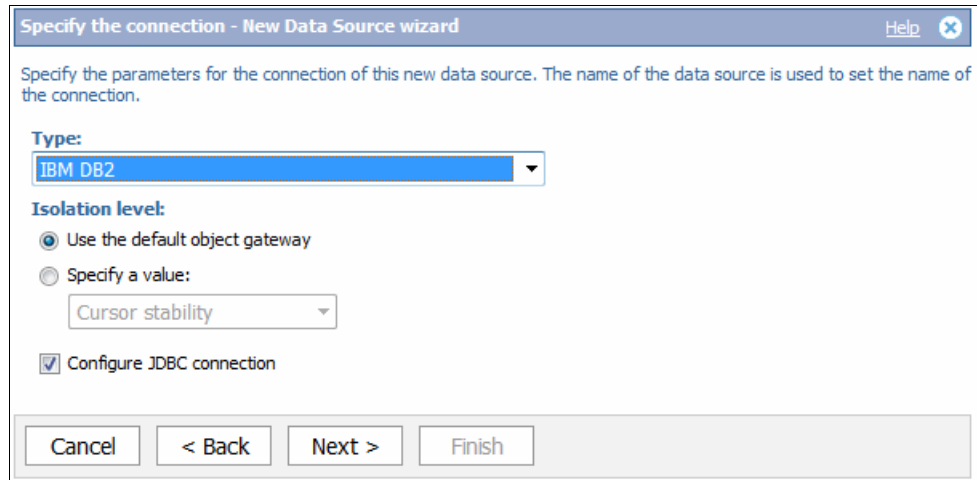


Figure 6-3 Selecting the data source type

4. Click **Next** to open the **Specify the IBM DB2 connection string** dialog (Figure 6-4 on page 172) and complete the following steps:
 - a. Enter Warranty claims in the DB2 database name field.
 - b. Check the **Open asynchronously** checkbox.
 - c. Select the **Signons** radio button.
 - d. Select **Password**, and then enter the user ID and password provided by your database administrator.
 - e. Click **Next** when done.

Specify the IBM DB2 connection string - New Data Source wizard Help X

Edit the parameters to build a DB2 connection string.

DB2 database name:

DB2 connect string:

Collation sequence:

Open asynchronously

Timeouts
 Specify the time in seconds, in which you want the database to connect or wait for your reply before timing out.

Connect time:

Reply time:

Signon
 Select whether or not authentication is needed, and if so, the type of authentication to use, whether a password is required and whether to create a signon.

No authentication

An external namespace:

Signons

Password

Create a signon that the Everyone group can use:

User ID:

Password:

Confirm password:

Testing
[Test the connection...](#)

Figure 6-4 Specifying the connection string to IBM DB2

Important testing note: You do not test the connection now because you have not yet configured Cognos to use the thick client interface for IBM DB2 and your test will generate errors. Further, a test at this point is not applicable for the Dynamic Query Mode capability you are configuring.

5. In the **Specify the IBM DB2 (JDBC) connection string** dialog (Figure 6-5):
 - a. Enter the server name or IP address for your DB2 server in the Server name field.
 - b. Change the value to match your server's port number in the Port number field.
 - c. Enter the database name as it is defined on the IBM DB2 Server in the Database name field.

Specify the IBM DB2 (JDBC) connection string - New Data Source wizard Help ✕

Edit the parameters to build a DB2 (driver: com.ibm.db2.jcc.DB2Driver) connection string.

Server name:
9.42.171.220

Port number:
50001

Database name:
ITSOWADB

JDBC Connection Parameters:
These optional parameters are appended to the URL and are specific to the driver.

Local Sort Options

Collation Sequence:

Level:
Primary ▾

Testing
[Test the connection...](#)

Cancel < Back Next > Finish

Figure 6-5 Specify the IBM DB2 JDBC connection string

- Click **Test the connection** to proceed, and then click **Test** to validate a successful connection (Figure 6-6)

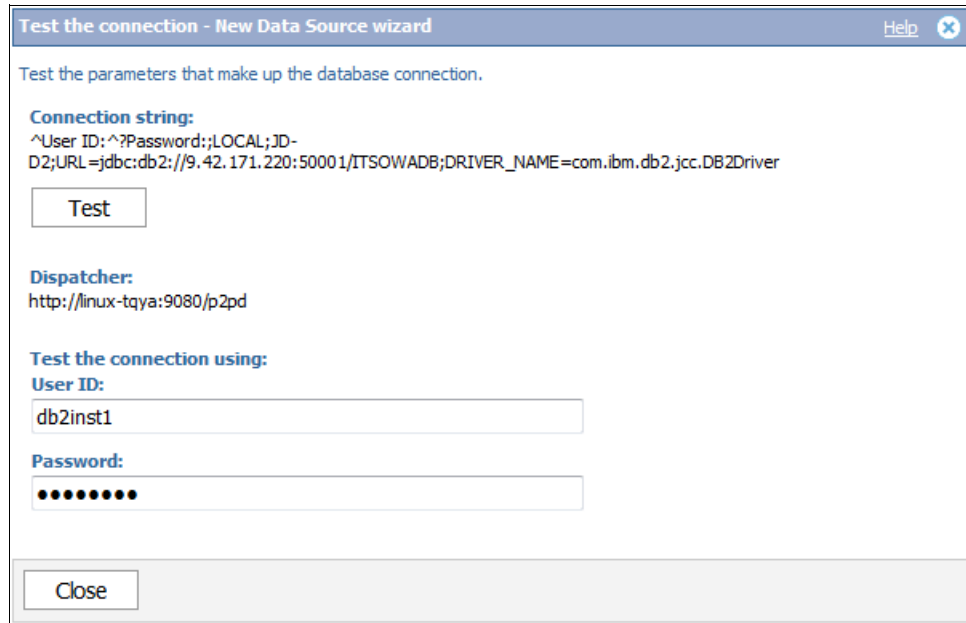


Figure 6-6 Test the connection

- If the connection is successful, a message will display to confirm that your test succeeded (Figure 6-7).

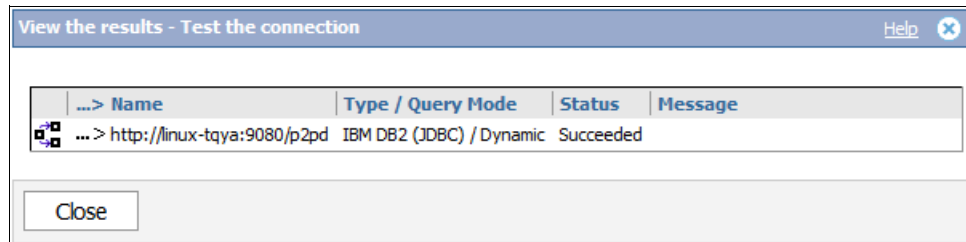


Figure 6-7 Successful connection message

- Click **Close** in the “View the results” dialog, and click **Close** again in the “Test the connection” dialog.
- Then click **Finish** in the “Specify the IBM DB2 (JDBC)” connection string window.

6.3 Creating the metadata model for reporting

Now that you have created your data source connection, you can create your metadata model within IBM Cognos Framework Manager.

To create the metadata model for reporting:

- Create a new model and import the source metadata.
- Model the metadata model.
- Publish the model to the server.

6.3.1 Create a new model and import the source metadata

You can import the metadata that you need for the warranty project using one of the following methods:

1. Import our metadata directly from the database.
2. Import the metadata from an exported model from IBM InfoSphere Data Architect.

Both methods work well, so choosing how you import your metadata comes down to personal preference. That said, importing metadata from IBM InfoSphere Data Architect IBM the Cognos Framework Manager model can reduce duplication in some cases, so you may want to opt for this method.

Importing metadata directly from the database

To begin:

1. Open IBM Cognos Framework Manager and click **Create a new project**.
2. In the New Project dialog (Figure 6-8):
 - a. Enter Warranty Tracking Program in the Project name field and choose an appropriate location to store your project.
 - b. Check the **Use Dynamic Query Mode** checkbox.
 - c. Click **OK** when done.

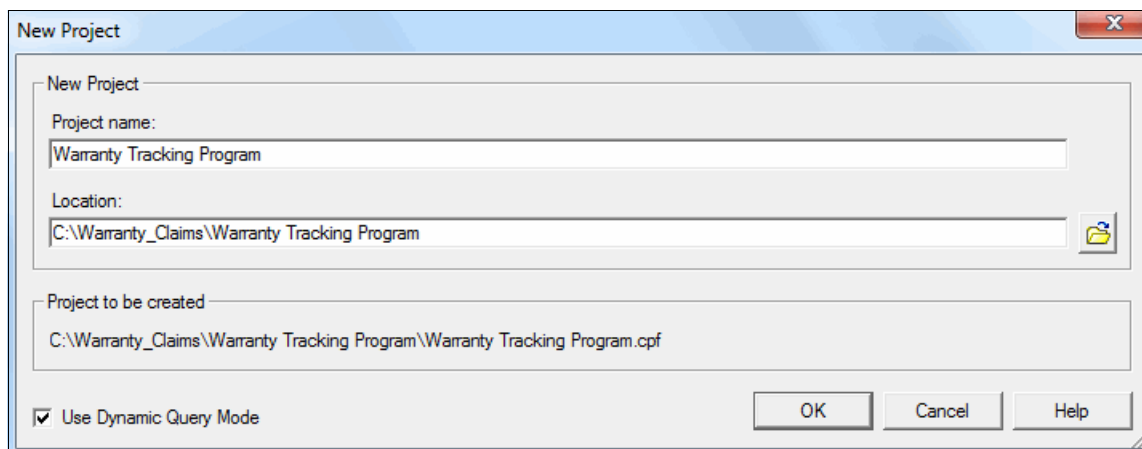


Figure 6-8 New project

Tip: If the directory that you specify does not exist, you can click **OK** to create it when prompted.

3. Choose **English** when prompted to select a language, and click **OK**.
4. Log in to IBM Cognos Business Intelligence, if prompted, using the credentials provided by your administrator.
5. The Metadata Wizard runs automatically, so select **Data Sources** from the list and click **Next**.
6. Select **Warranty claims database** from the list, and click **Next**.

A list of available objects in the database displays. You import only the desired objects that will make up this application. In this example, the objects that you want to import are under the WARRANTY schema.

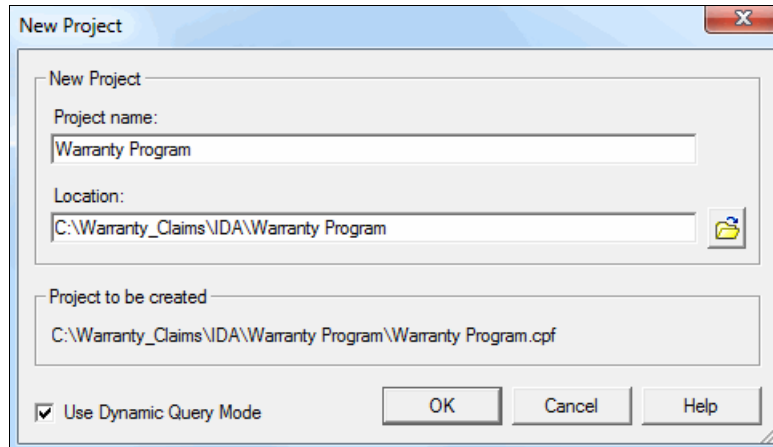


Figure 6-10 Create a new project

Tip: If the directory that you specify does not exist, you can click **OK** to create it when prompted

3. Choose **English** when prompted to select a language, and click **OK**.
4. Select **IBM Cognos Model** from the list on the Select Metadata Source dialog.
5. Browse to the location of the exported model from IBM InfoSphere Data Architect and open the file.

Tip: The exported model might not have an extension, so you may need to perform a wildcard search for the extension asterisk (*.*) to locate the file in the **Files of type** drop-down menu.

6. Click **Next** on the Select Model dialog.
7. In the Select Objects dialog:
 - a. Select the ITSOWADB top level Project Namespace.
 - b. Deselect the CUSTOMER PRODUCT SALES database query subject and Subject Area namespace as shown in Figure 6-11 on page 178.
 - c. Click the **Import and create a new name** radio button.
 - d. Click **Next**.

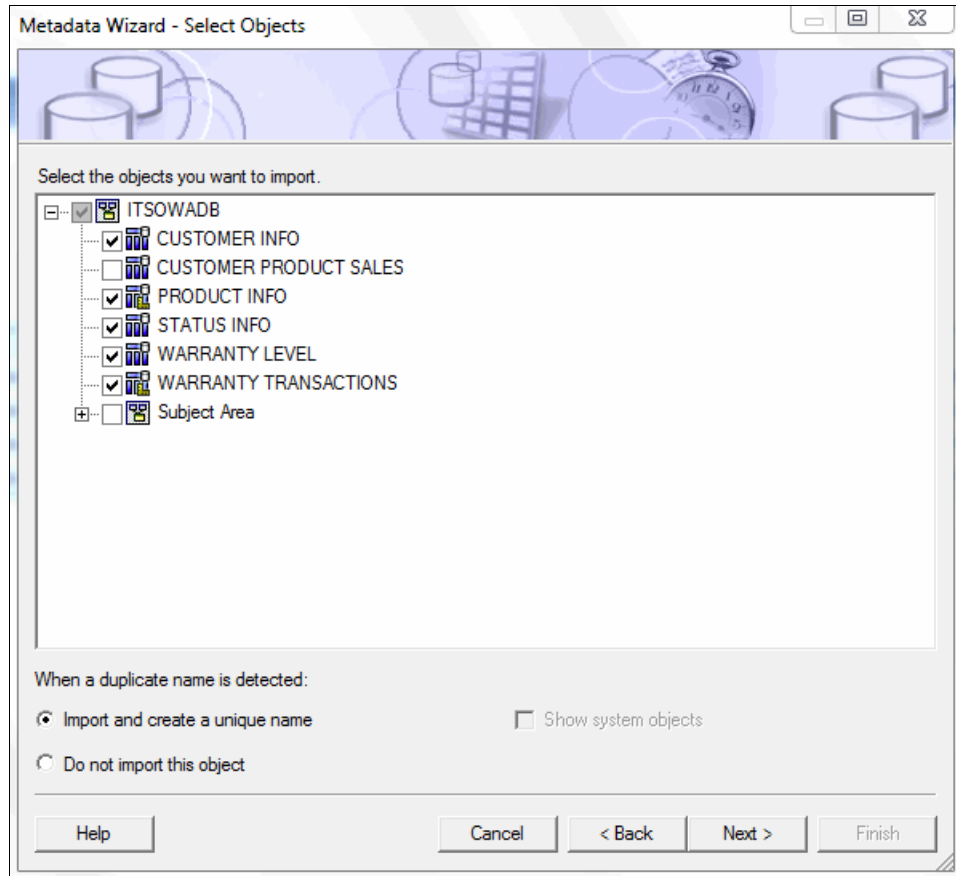


Figure 6-11 Object selection

- IBM Cognos Framework Manager imports the objects selected from your model and removes all that were not enabled. Figure 6-12 shows the results from the import.

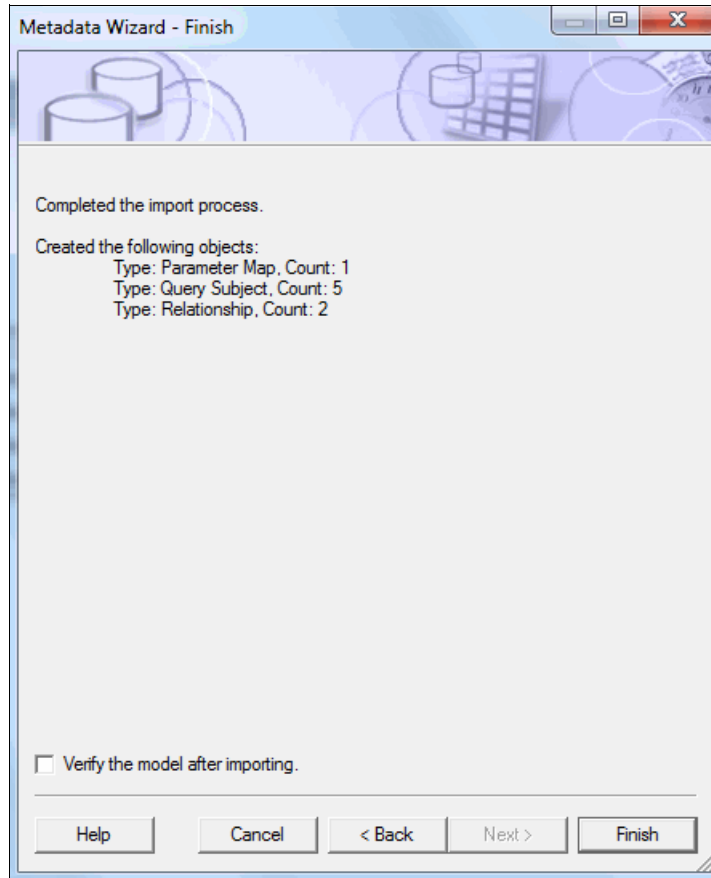


Figure 6-12 Import results

- Click **Finish**.
- Now you need to confirm that you have a working starting point. In the Project Viewer, on the left side of the IBM Cognos Framework Manager user interface, expand the Data Sources entry by clicking the plus sign (+) beside the folder icon, as shown in Figure 6-13.

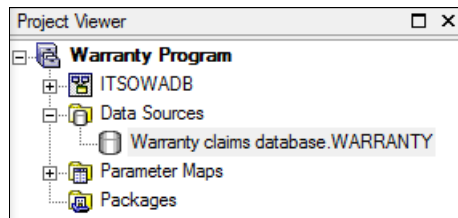


Figure 6-13 Project viewer

- Right-click the data source reference and select **Test**. You should receive a message that confirms that your data source was tested successfully.

Tip: If you receive an error message indicating that the test was unable to find the data source in the content store, correct the name you typed in the Content manager Data Source properties to align it with the one in the content store.

12. In the Project View, expand the ITSOWADB entry by clicking the plus sign (+).
13. Right-click and select each database query subject under the ITSOWADB entry to highlight all the items. Then, from the tools menu, select **Tools** → **Validate Object**.

Validation errors: A successful validation without issues will yield no results delivered back to the user. If there are errors, they will be visible.

6.3.2 Model the metadata model

The imported metadata will serve as the foundation for our warranty sample report packages.

The left pane of the IBM Cognos Framework Manager application shows the Project Viewer, which includes a tree structure of our warranty tracking model (Figure 6-14).

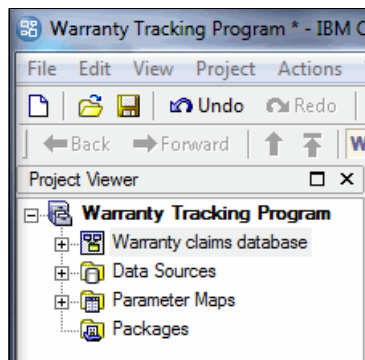


Figure 6-14 Warranty Tracking Program Model

The model starts with the Warranty Tracking Program project, which consists of the Warrant claims database namespace and containers for Data Sources, Parameter Maps, and Packages containers. The work for the warranty sample resides in the Warranty claims database namespace and Packages container. To begin your model, expand the Warranty claims database namespace and build the database view.

Building the database view

The database view stores the original imported objects from your source systems. Here, you can make minor corrections to data relationships and object usage because these corrections will be reflected in the queries that IBM Cognos Dynamic Query creates. They will also minimize further impact later.

To create a database view:

1. Right-click the **Warranty claims database** namespace and select **Create** → **Namespace**.
2. Name your namespace *Database view*, or something meaningful.
3. Next, repeat steps 1 and 2 to create two new namespaces entitled: *Business view* and *Publish view* (Figure 6-15 on page 181).

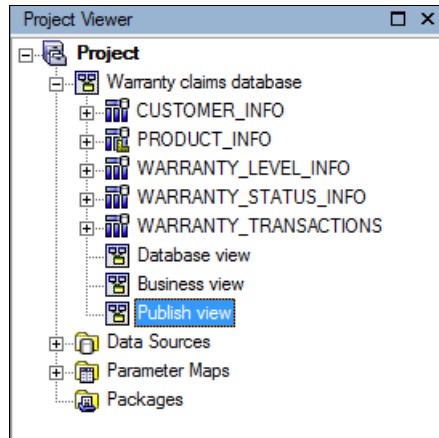


Figure 6-15 Create the Database, Business, and Publish views

4. Select all of the non-namespace objects (CUSTOMER_INFO through WARRANTY_TRANSACTIONS) under the Warranty claims database namespace, right-click and select **Edit** → **Cut**.
5. Right-click the Database view namespace and select **Edit** → **Paste** to populate your database tables as shown in Figure 6-16.

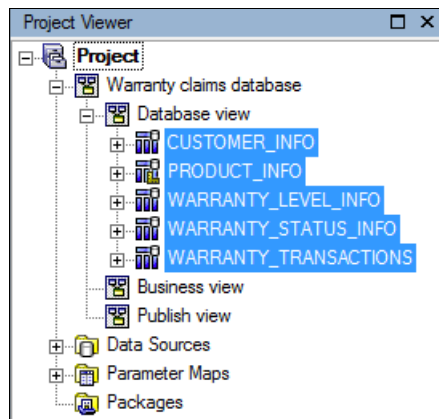


Figure 6-16 Database view

6. The only action you perform directly in the Database view namespace is to correct any issues with relationships or data types. If you expand these Database tables, you will be able to see the associated data columns.

- Hold down the Ctrl key and click the **PRODUCT_VALUE**, **CUSTOMER ID**, **PROD_SER_NUM**, **WARRANTY LEVEL ID**, and **STATUS ID** columns to highlight all as shown in Figure 6-17; note that all have the same icon, a yellow letter L

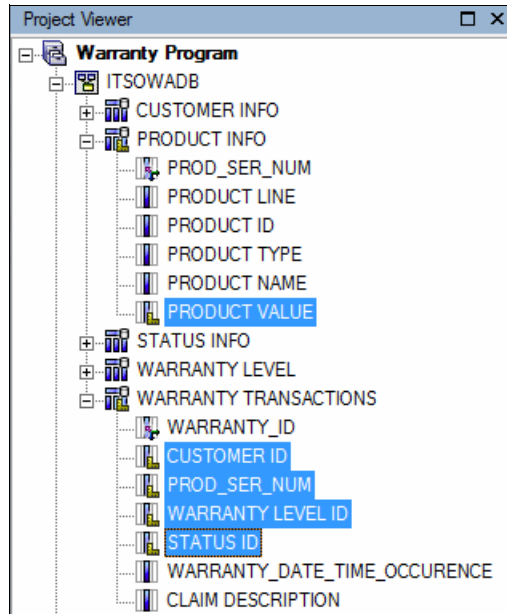


Figure 6-17 Selecting columns

- Locate the Usage column using the bottom scroll bar.
All of these items are listed as *Facts*, which means these columns are assumed to contain data that can be aggregated in some way. That said, the data in these columns cannot be aggregated for any useful information because it consists of attributes and identifiers. *Attributes* are columns of information about a given record that should not be aggregated, while *Identifiers* are used to represent primary or foreign keys for use in relationships between tables. You need to change the property value for each.
- Change the Usage property of PRODUCT VALUE from *Fact* to *Attribute*.
- Change the Usage property of CUSTOMER_ID, PROD_SER_NUM, WARRANTY LEVEL ID, and STATUS ID from *Fact* to *Identifier* (Figure 6-18).

Properties	
Properties	Language
	Usage ▼
CUSTOMER ID	Identifier
PROD_SER_NUM	Identifier
WARRANTY LEVEL ID	Identifier
STATUS ID	Identifier
PRODUCT VALUE	Attribute

Figure 6-18 Changing the usage properties

11. In the Project Viewer, right-click the Database view namespace and select **Switch to View** → **Diagram** (Figure 6-19).

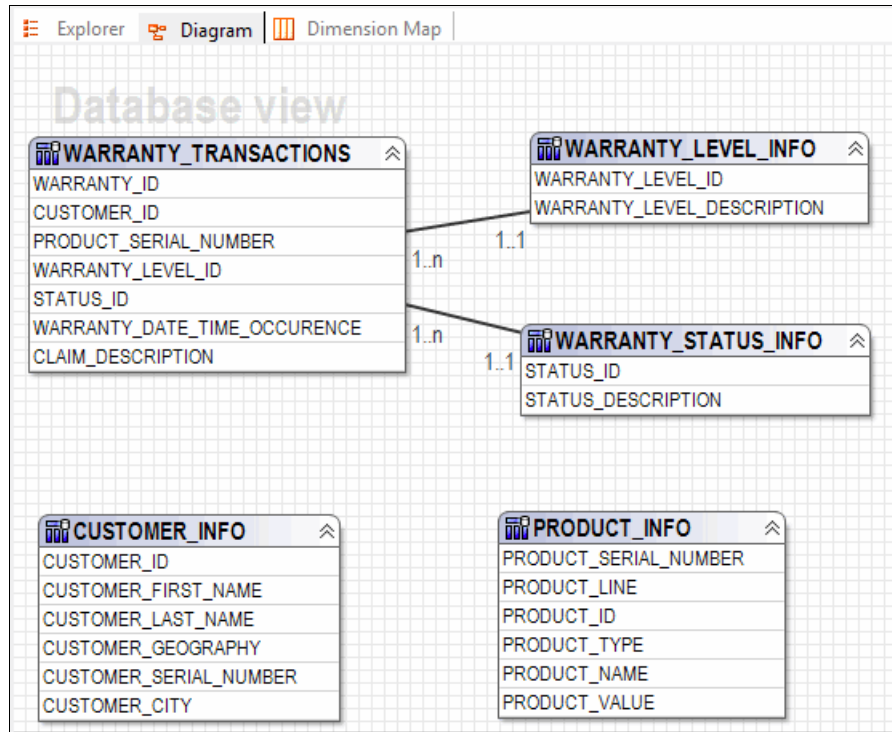


Figure 6-19 Diagram view

Tip: The view should show the five tables and the two relationships. If these tables and relationships are not visible, access the Diagram menu at the top of the Framework Manager dialog, click the **Auto Layout** icon, accept the defaults, then click **Apply** followed by **Close** to view the five tables and the two relationships.

Since you did not import the CUSTOMER_SALES_INFO table, you need to create relationships to access data from both the CUSTOMER_INFO and PRODUCT_INFO tables.

1. Hold down the Ctrl key, click the top of the WARRANTY_TRANSACTIONS table and then click the top of the CUSTOMER_INFO table.
2. Next, right-click the top of the CUSTOMER_INFO table and select **Create** → **Relationship**.

The Relationship Definition dialog opens to allow you to define how the tables are related.

1. Ensure that the Query subject for both tables is CUSTOMER_ID and that the Cardinality for WARRANTY_TRANSACTIONS is 1..n (that is, one-to-many) and 1..1 (one-to-one) for CUSTOMER_INFO as shown in Figure 6-20 on page 184; click **OK** to confirm these settings.

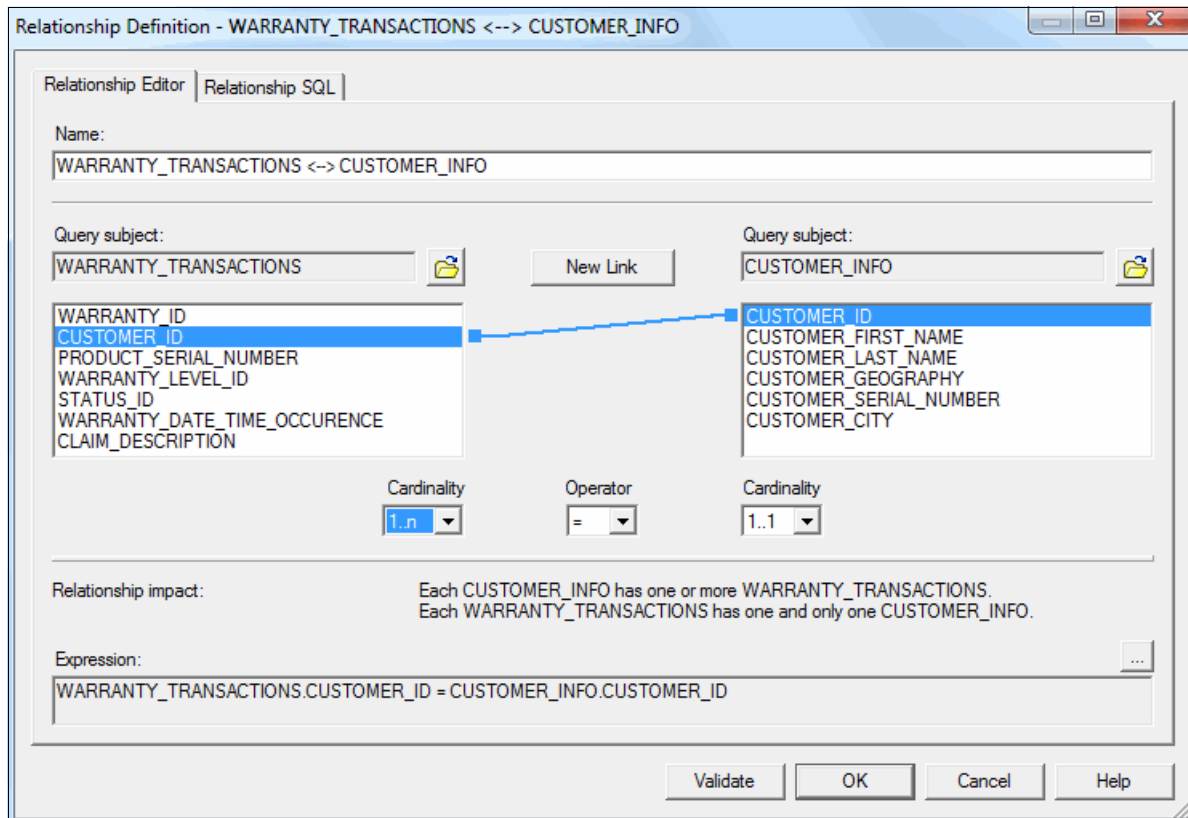


Figure 6-20 CUSTOMER_INFO relationship definition

- Next, hold down the Ctrl key, click **PRODUCT_SERIAL_NUMBER** in the WARRANTY_TRANSACTIONS table, and click **PRODUCT_SERIAL_NUMBER** in the PRODUCT_INFO table, and then right-click **PRODUCT_SERIAL_NUMBER** in the PRODUCT_INFO table and select **Create** → **Relationship**.

- Ensure that the **Query subject** for both tables is **PRODUCT_SERIAL_NUMBER** highlighted and the Cardinality for WARRANTY_TRANSACTIONS side is 1.n and 1.1 for the PRODUCT_INFO as shown in Figure 6-21; click **OK** to confirm these settings.

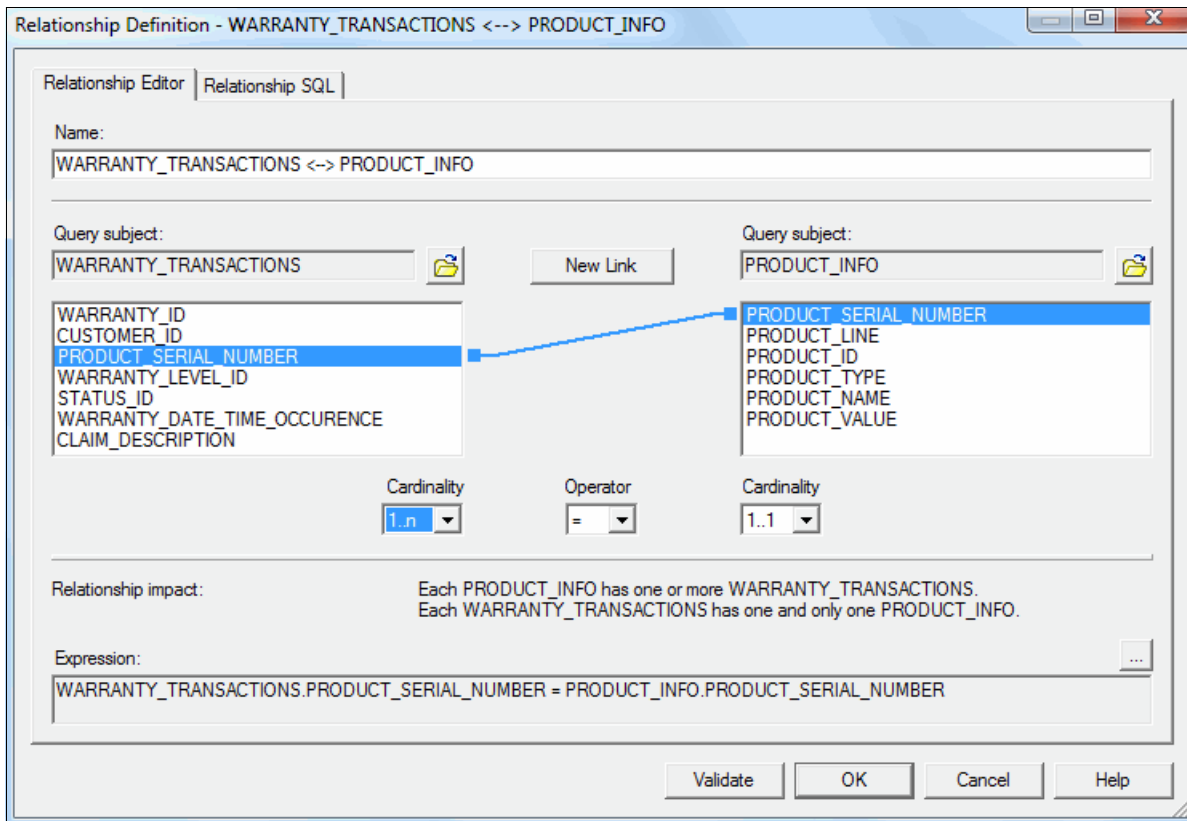


Figure 6-21 PRODUCT_INFO relationship definition

Your diagram should now look as shown in Figure 6-22 on page 186.

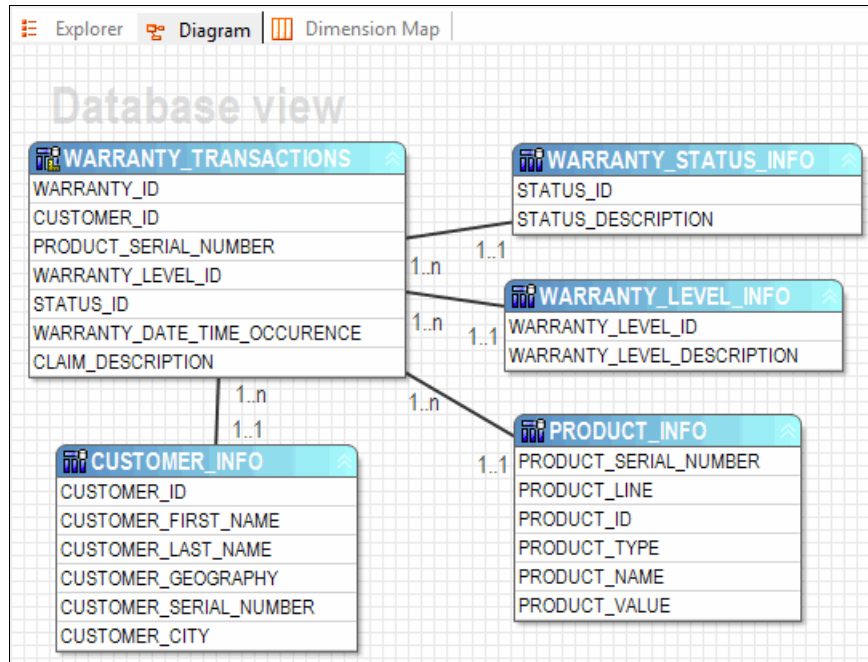


Figure 6-22 Resulting diagram

Tip: If you imported using the model from IBM InfoSphere Data Architect, the original two relationships on the WARRANTY TRANSACTIONS table might show a cardinality of 0..n. You can change this value to 1..n by double-clicking the relationship line, changing the value on the WARRANTY TRANSACTIONS side to 1..n and clicking **OK** to accept the change.

4. Double-click the Project label in the Project Viewer to return to the main project view.

Building the Business view

Now adjust items in the Business view to make it easier for your users to find and understand the data in the model. Basically, you are using business concepts to model your business view. This view is where you can apply business rules or security, as well. You will be creating and testing the following query subjects for your Business view:

- ▶ Customer Information
- ▶ Product Information
- ▶ Claim Status
- ▶ Claim Reason
- ▶ Warranty Transactions

A query subject is a set of query items that have an inherent relationship, and you modify them to optimize and customize the data that they retrieve.

In Project Viewer:

1. Right-click the Business view namespace and select **Create** → **Query Subject**.
2. In the New Query Subject - Name and Type dialog, enter Customer Information in the Name field and click **OK**. In the Query Subject Definition dialog, click the plus sign (+) beside the Database view namespace in the Available Model Objects pane followed by the plus sign (+) beside the CUSTOMER_INFO table.

3. Drag all the children objects (columns) from the CUSTOMER_INFO table to the Query Items and Calculations pane (Figure 6-23), and click **OK** when done.

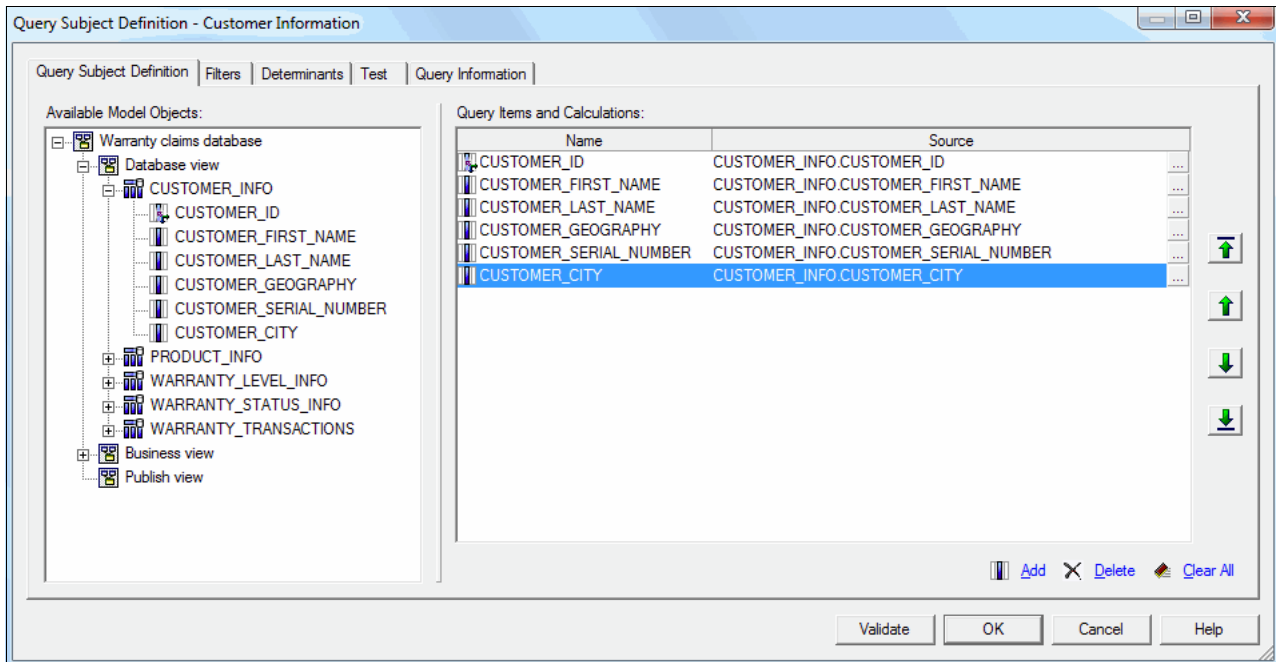


Figure 6-23 Customer Information query subject definition

4. Right-click **Customer Information** and select **Test** to open the Test Results dialog.
5. Click **Test Sample** to test the query subject.
6. After reviewing the displayed result set (Figure 6-24), click **Close**.

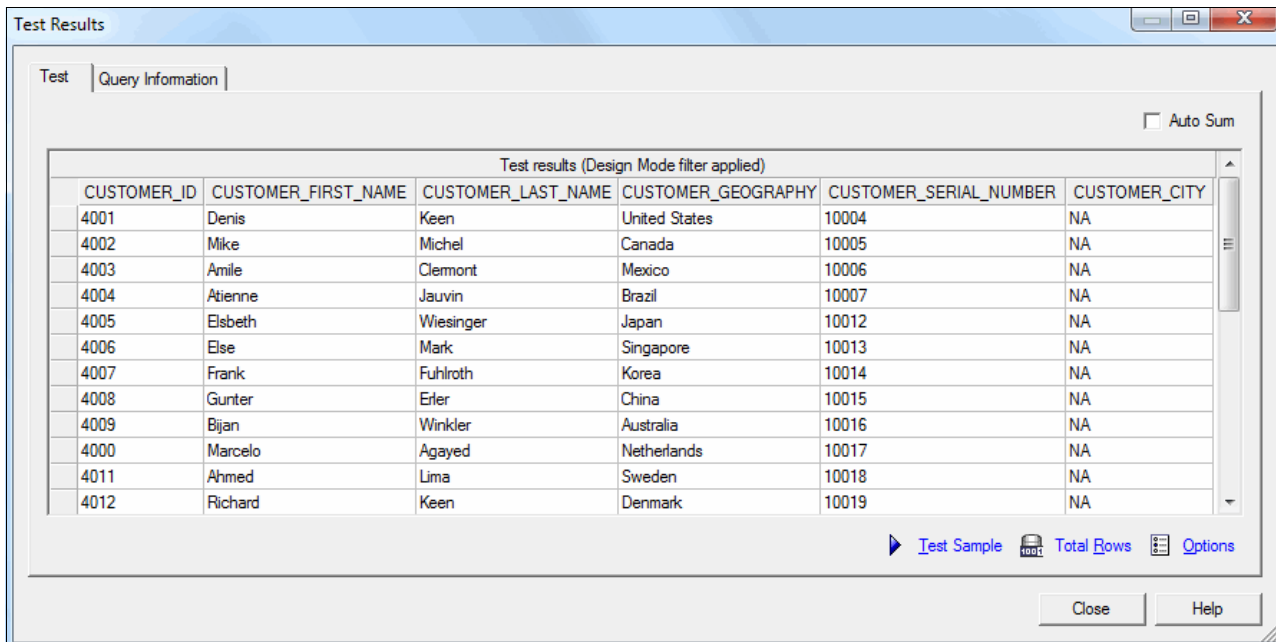


Figure 6-24 Customer Information Query Subject Test results

Now repeat these steps to create and test the Product Information query subject. We have highlighted only those steps unique to this query subject.

1. In the New Query Subject - Name and Type dialog, enter Product Information in the Name field and click **OK**. In the Query Subject Definition dialog, click the plus sign (+) beside the Database view namespace in the Available Model Objects pane followed by the plus sign (+) beside the table PRODUCT_INFO.
2. Drag all the children objects (columns) from the PRODUCT_INFO table to the Query Items and Calculations pane (Figure 6-25) and click **OK** when done.

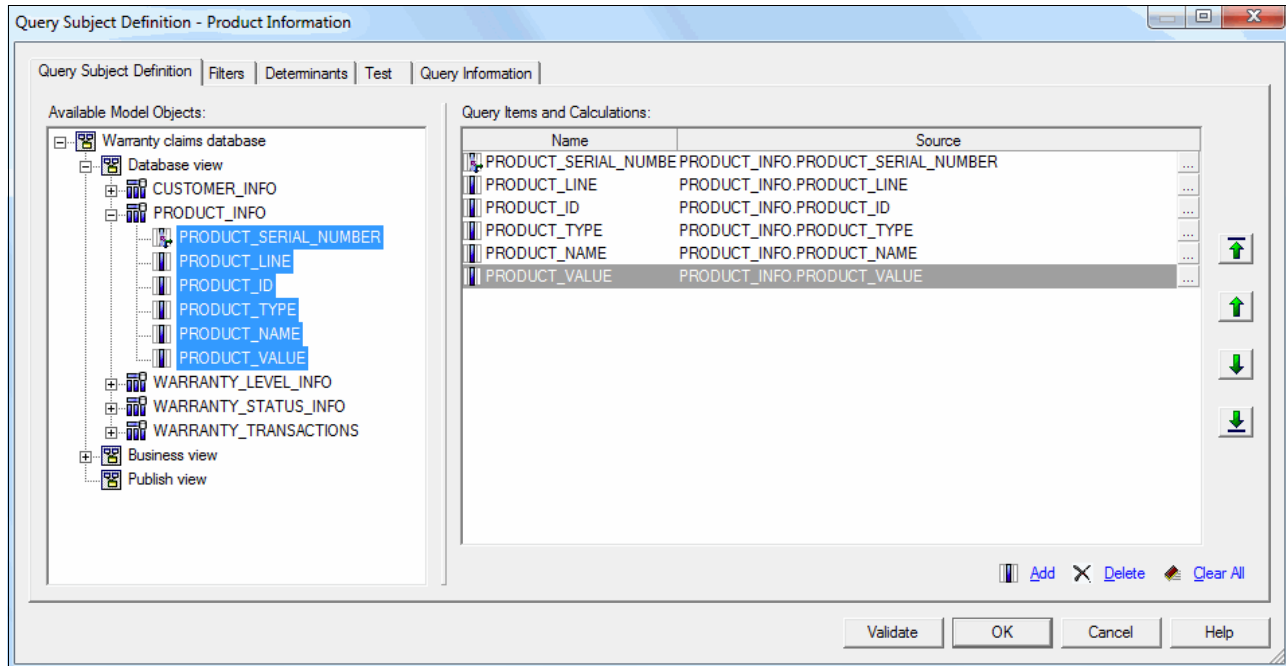


Figure 6-25 Product Information Query subject definition

3. Run the Test Sample, review the results and close the Test results dialog when done (Figure 6-26).

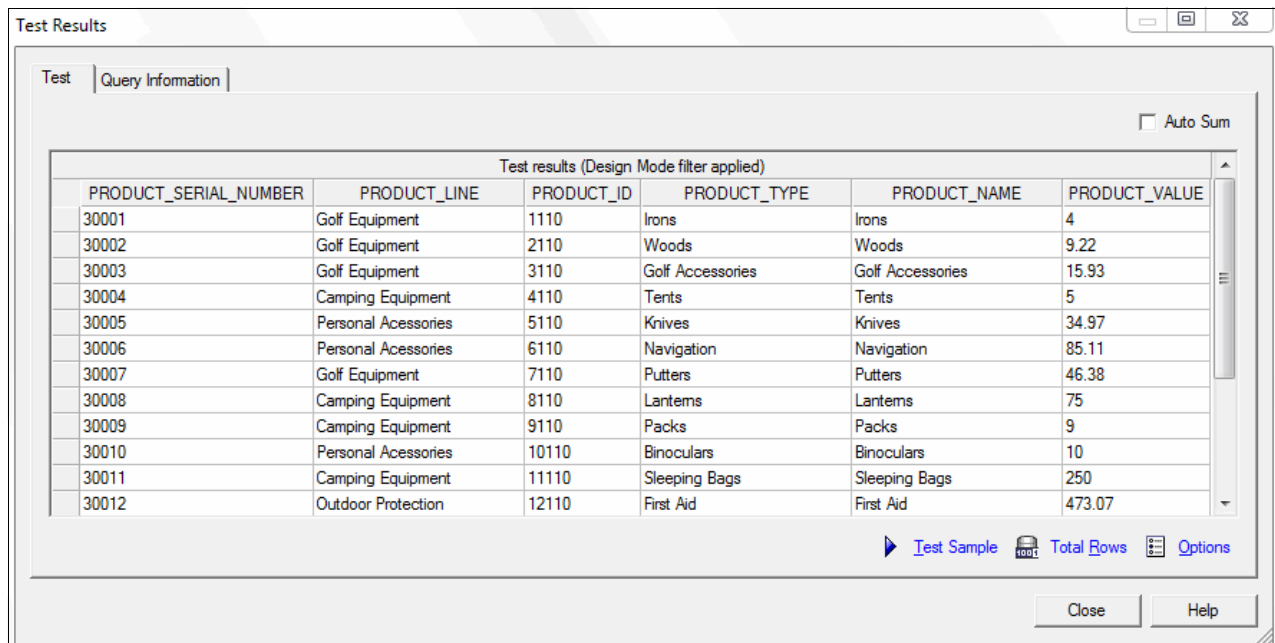


Figure 6-26 Product Information Query Subject Test results

We've highlighted unique actions for creating and testing the Claim Status query subject as in Figure 6-26 on page 188:

1. In the New Query Subject - Name and Type dialog, enter Claim Status in the Name field and click **OK**.
2. In the Query Subject Definition window, click the plus sign (+) beside the Database view namespace in the Available Model Objects pane followed by the plus sign (+) beside the table WARRANTY_STATUS_INFO.
3. Drag the children objects (columns) from the WARRANTY_STATUS_INFO table to the Query Items and Calculations pane (Figure 6-27) and click **OK** when done.

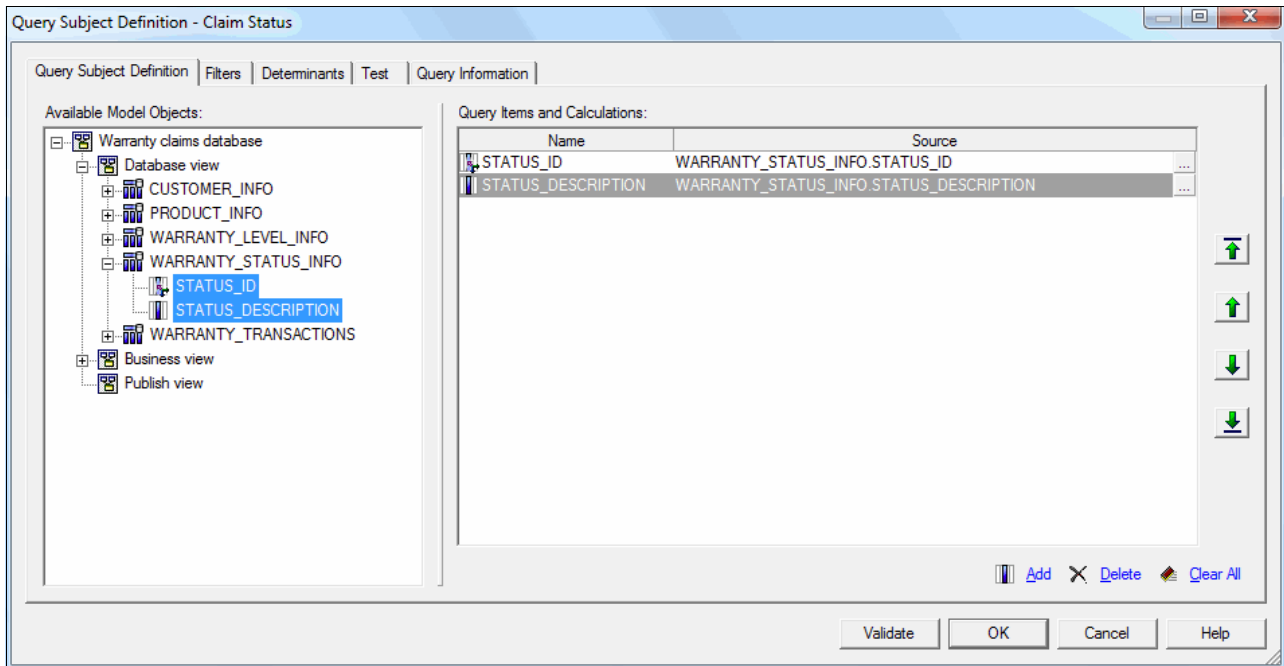


Figure 6-27 Claim Status Query subject definition

4. Run the Test Sample, review the results and close the Test results dialog when done (Figure 6-28 on page 190).

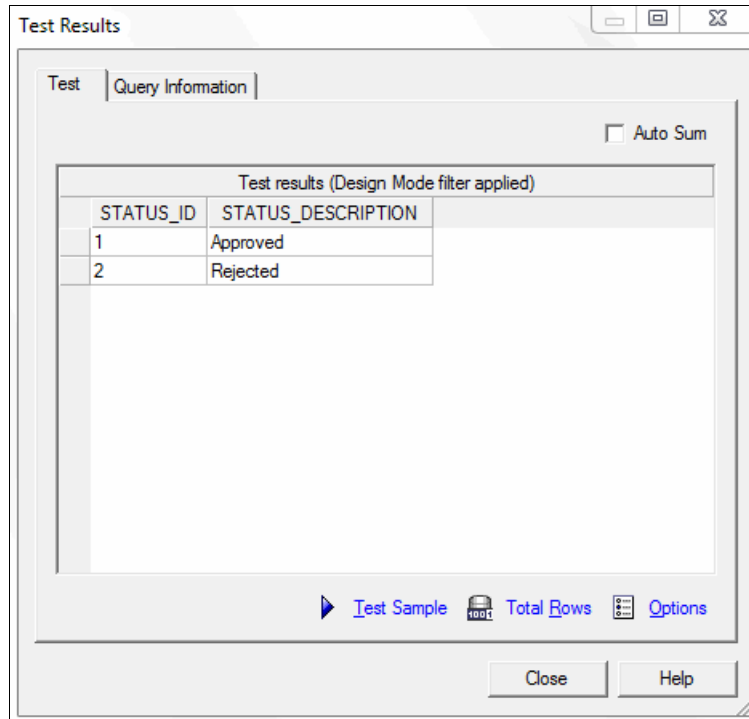


Figure 6-28 Claim StatusTest results

Following are the unique actions for creating and testing the Claim Status query subject:

1. In the New Query Subject - Name and Type window, enter Claim Reason in the Name field, and click **OK**.
2. In the Query Subject Definition window, click the plus sign (+) beside the Database view namespace in the Available Model Objects pane followed by the plus sign (+) beside the table WARRANTY_LEVEL_INFO.
3. Drag the children objects (columns) from the WARRANTY_LEVEL table to the Query Items and Calculations pane (Figure 6-29 on page 191) and click **OK** when done.

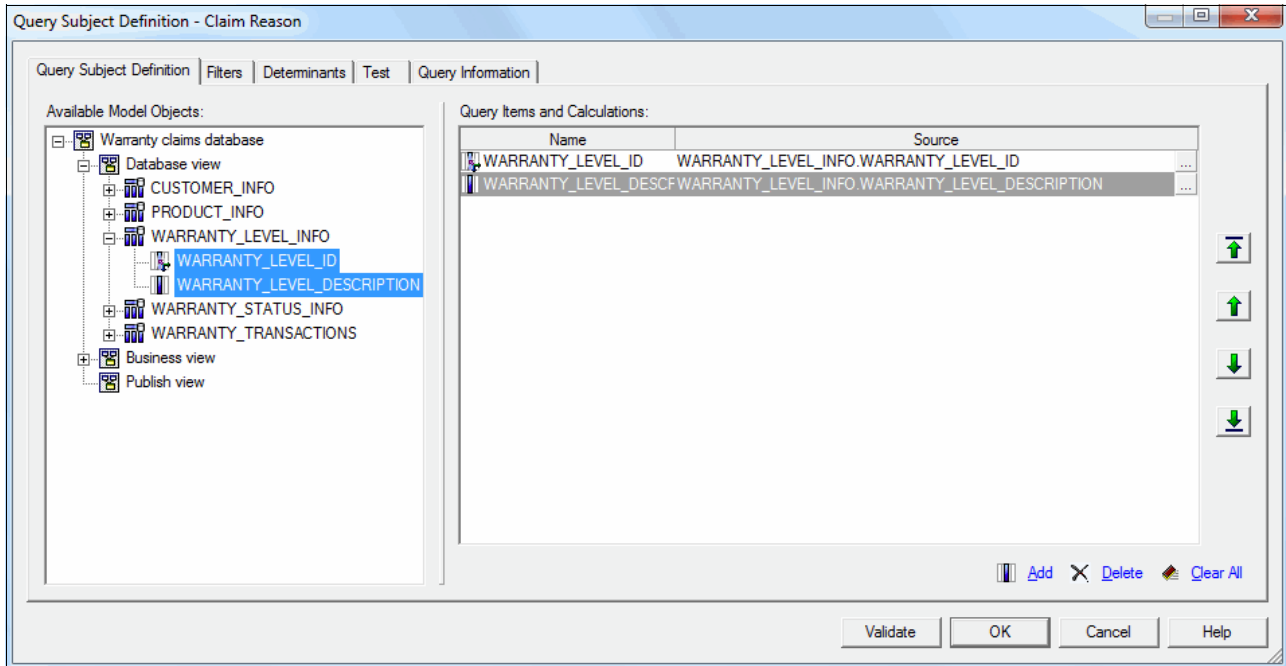


Figure 6-29 Claim Reason Query subject definition

4. Run the Test Sample, review the results and close the Test Results dialog (Figure 6-30) when done.

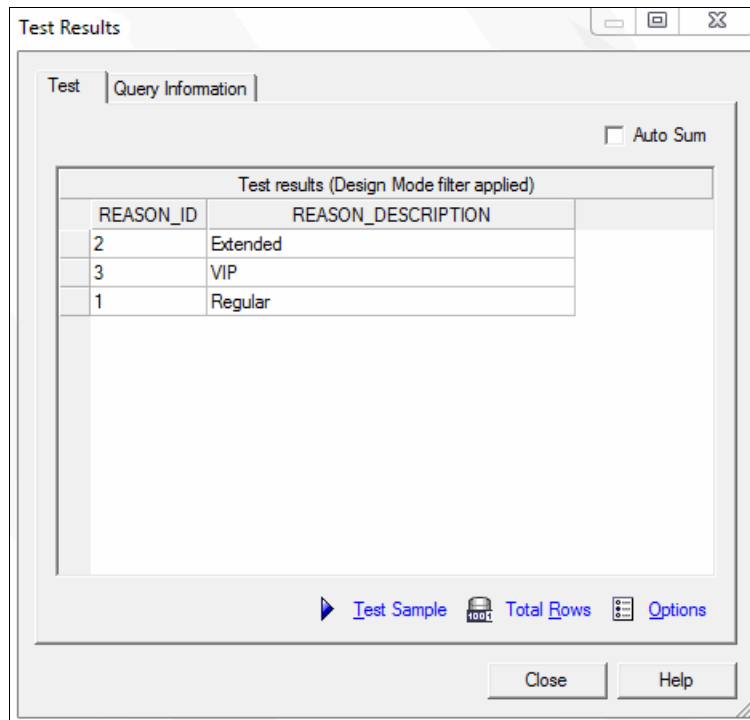


Figure 6-30 Claim Reason Test results

We have highlighted unique actions for creating and testing the Claim Status query subject:

1. In the New Query Subject - Name and Type dialog, enter Warranty Transactions in the Name field and click **OK**.

- In the Query Subject Definition window, click the plus sign (+) beside the Database view namespace in the Available Model Objects pane followed by the plus sign (+) beside the table WARRANTY_TRANSACTIONS.
- Drag the children objects (columns) of the WARRANTY_TRANSACTIONS table into the Query Items and Calculations pane (Figure 6-31) and click **OK** when done.

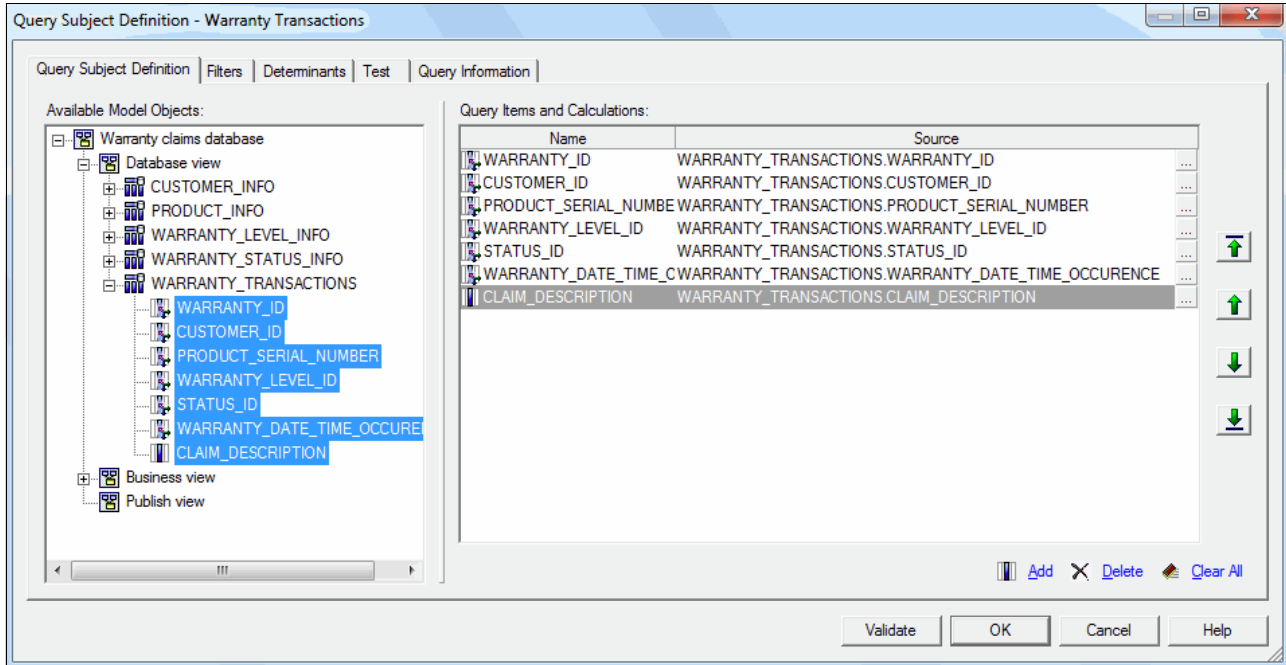


Figure 6-31 Warranty Transactions Query subject definition

- Run the Test Sample, review the results and close the Test Results dialog (Figure 6-32) when done.

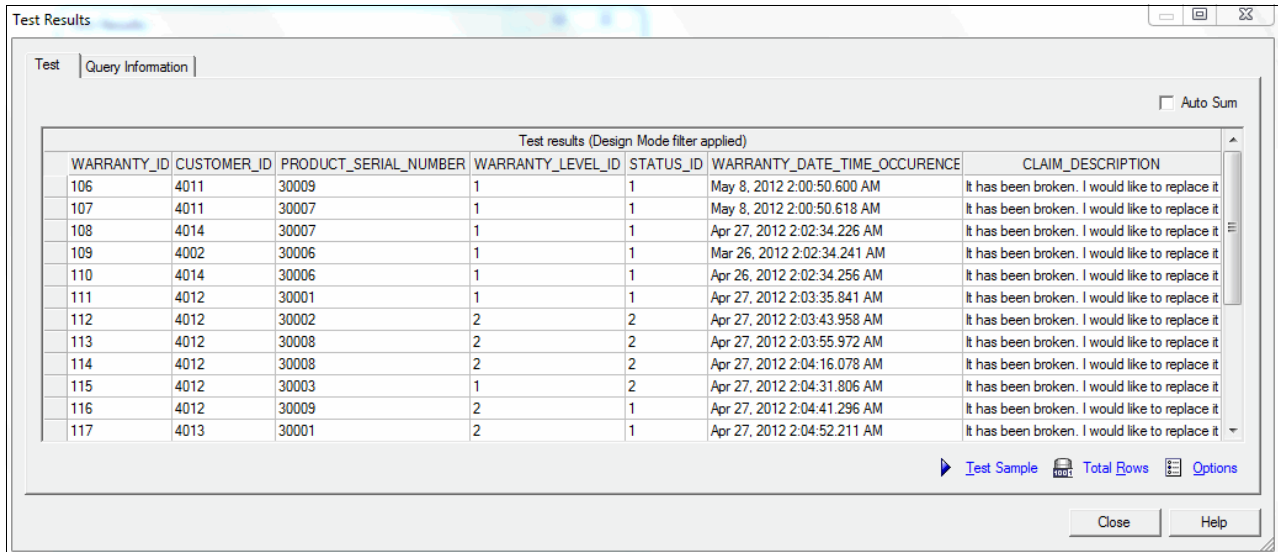


Figure 6-32 Warranty Transactions Test results

Now that you have created the model query subjects, you can rename these objects.

1. With the Business view expanded in the Project Viewer, click the plus sign (+) to expand the Claim Reason query subject, and then rename the following objects:
 - Click the **WARRANTY_LEVEL_ID** query item, and in the Properties window, edit the Name field to read REASON_ID; press Enter when done.
 - Click the **WARRANTY_LEVEL_DESCRIPTION** query item, and in the Properties window, edit the Name field to read REASON_DESCRIPTION; press Enter when done.

Figure 6-33 shows the results of the renamed items in the Project Viewer.

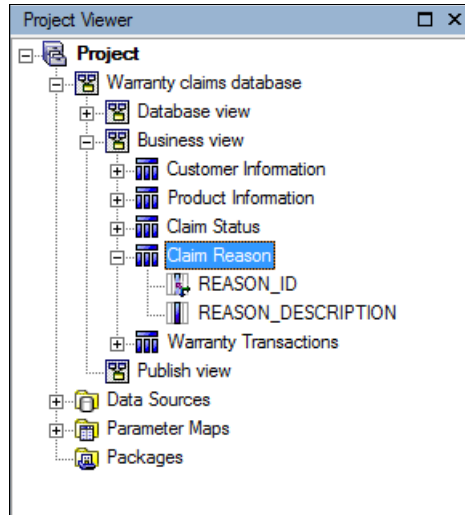


Figure 6-33 Renaming objects

2. Close the Claim Reason query subject in the Project viewer by clicking the minus (-) sign.
3. Expand the Warranty Transactions query subject in the Project Viewer Pane and highlight all the query items under the Warranty Transactions query subject.
4. Locate the Is Hidden column in the Properties pane, and set to **True** for CUSTOMER_ID, PRODUCT_SERIAL_NUMBER, WARRANTY_LEVEL_ID, STATUS ID and CLAIM_DESCRIPTION (Figure 6-34).

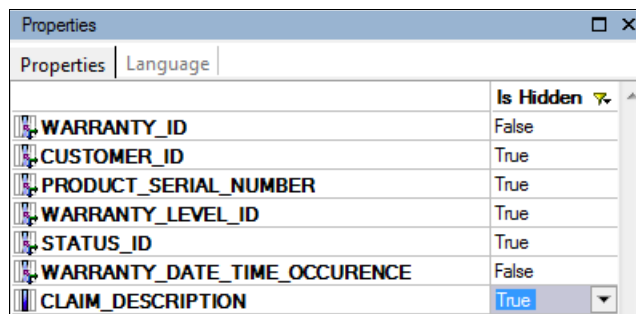


Figure 6-34 Is Hidden to true

5. Save the project by clicking **Save** from the File menu.

Creating the Publish view

The Publish view is the reference to objects in the Business view that you might want to expose to report authors or consumers via a package. Only objects that might get published should be added to this view.

To create the Publish view:

1. Highlight all the Query Subjects under the Business view in the Project Viewer window, right-click any of the subjects and select **Edit** → **Copy**.
2. Click the Publish view namespace in the Project Viewer, right-click the Publish view and select **Edit** → **Paste Shortcut**.
3. Rename each of the items now under the Publish view to remove the shortcut to text (Figure 6-35).

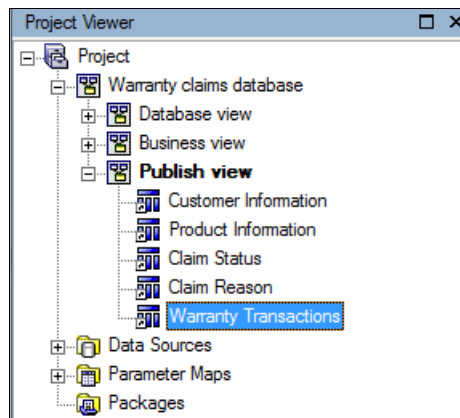


Figure 6-35 Publish view

6.3.3 Publish the model to the server

You need to publish your package to allow users to access the data from the web. IBM Cognos Business Intelligence provides a wizard for you to publish your packages.

You can have many package definitions that allow you to tailor each package to meet the needs of specific users or lines of business. Remember, your objective is to enable users to be able to find the data that they require on their own. To publish your model to the server:

1. Right-click the packages in the Project Viewer and select **Create** → **Package**. In the Create Package dialog, enter Warranty Tracking in the Name field and click **Next**.
2. In the Define objects dialog, deselect the Database view and the Business view, leaving only the Publish view selected (Figure 6-36 on page 195) and click **Finish** when done.

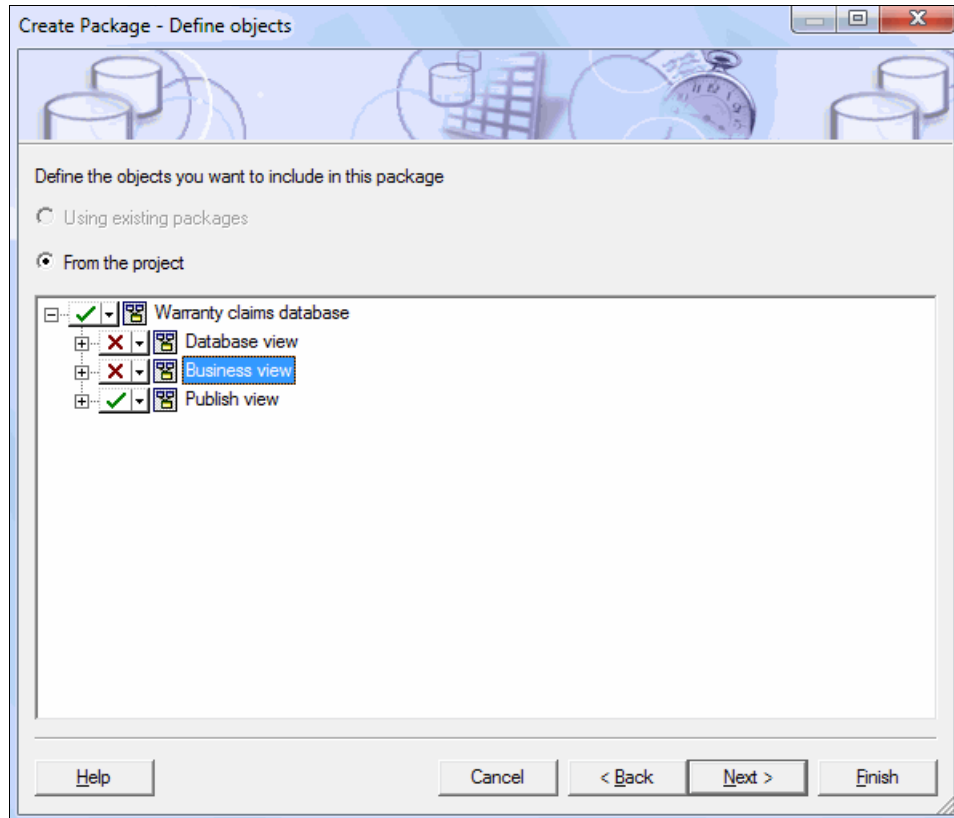


Figure 6-36 Select the Publish view only

3. Select **Yes** when prompted to open the Publish Package wizard.
4. Accept the defaults in the Select Location Prompt by clicking **Next**.
5. Accept the defaults in the Add Security prompt by clicking **Next**.

6. On the Options page, keep the defaults, and click **Publish**. Then, click **Finish** (Figure 6-37).

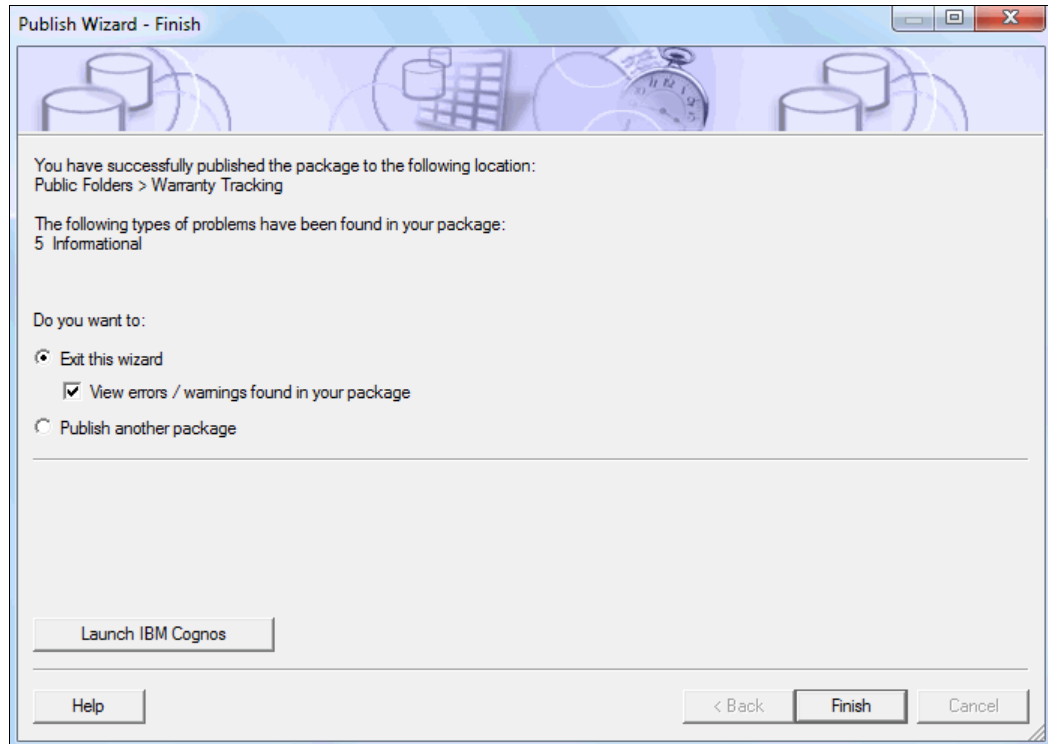


Figure 6-37 Publish wizard

7. Review the messages from the model verification task, which are all informational, and click **Close** when done.

6.4 Authoring reports with IBM Cognos Report Studio

IBM Cognos Report Studio is a robust report design and authoring tool. Using IBM Cognos Report Studio, report authors can create, edit, and distribute a wide range of professional reports. They can also define corporate-standard report templates for use in IBM Cognos Query Studio and edit and modify reports created in IBM Cognos Query Studio or IBM Cognos Analysis Studio.

This tool enables:

- ▶ Professional report authors to build sophisticated, multiple-page, multiple-query reports against multiple databases.
- ▶ Business users to create ad-hoc reports and dashboards.
- ▶ Mobile users to access and view reports from mobile devices.

Selecting the most suitable way to show the data to users is an important task for a Cognos report developer. Cognos Report Studio offers the following report types to help developers design the reports in an effective way:

- ▶ List reports
- ▶ Crosstabs
- ▶ Charts (such as pie, bar, gauge, quadrant, map, and other types of charts)

- ▶ Active reports
- ▶ Financial reports
- ▶ Statistics reports (which requires the installation of a statistical analysis package)

In addition to these tools, Cognos Report Studio can manipulate data such as focus reports using prompts, filter data using filters, drill through from one report to another report, distribute reports through bursting, and enhance the report layout.

For more information about Cognos Report Studio, see the Report Studio User Guide 10.1.1 at:

http://public.dhe.ibm.com/software/data/cognos/documentation/docs/en/10.1.1/ug_cr_rptstd.pdf

Based on the requirements for our warranty sample, all reporting must be developed with the company logo, time, page number, and the report name. For usability reasons, our report author creates a custom template that can be used every time a new report is created so that all the reports will have a consistent layout.

Creating the report template

A report template is a pattern report authors use to build reports. It is good practice to create your own report templates when you frequently produce the same type of reporting. Using templates will allow you to ensure consistency for all your reporting.

Before you continue: It is assumed for the remainder of the chapter that the IBM Cognos Business Intelligence 10.1.1 server is up and running, which enables you to use Report Studio. Should you need more information about how to install and configure IBM Cognos Business Intelligence 10.1.1, see:

<http://publib.boulder.ibm.com/infocenter/cbi/v10r1m1/index.jsp>

To create a report template:

1. From the IBM Cognos Connection page, click **Launch** → **Report Studio** as shown in Figure 6-38.

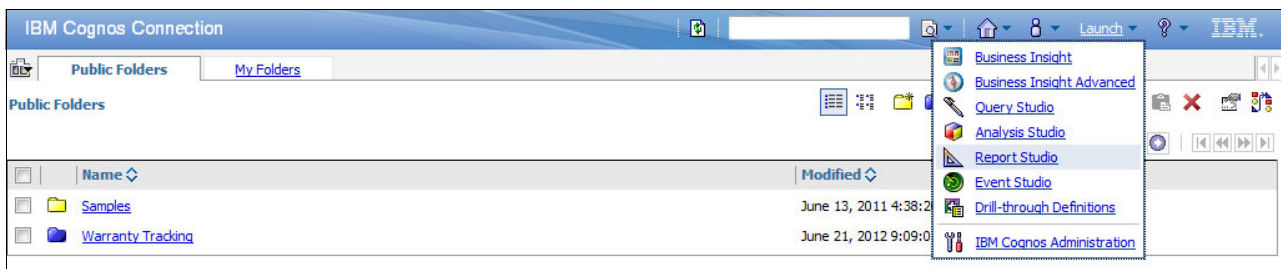


Figure 6-38 IBM Cognos Connection

2. Click **Warranty Tracking** from the “List of all packages” (Figure 6-39).

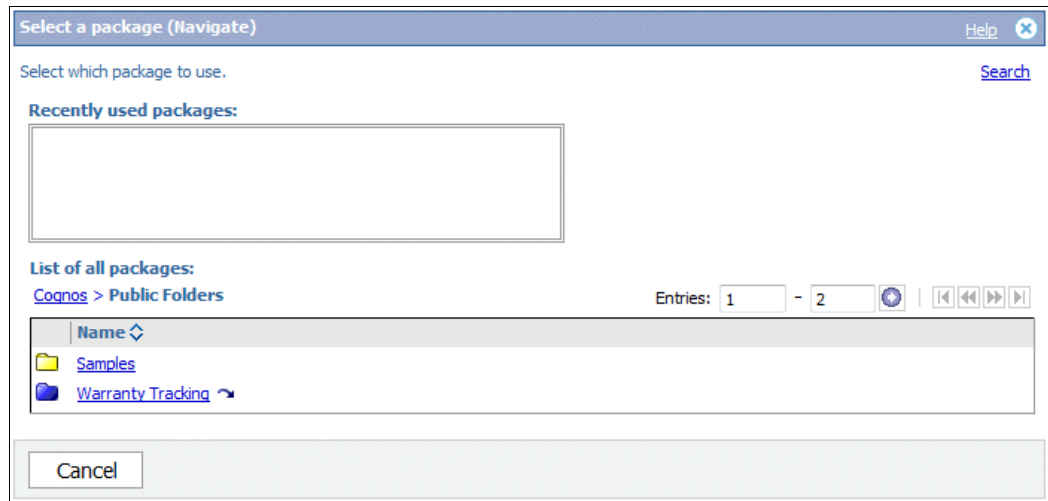


Figure 6-39 Package selection

3. Next, click **Create new** → **blank** → **OK**.
4. In the toolbar, click **Headers & Footers** followed by **Page Header and Footer**, and then select both **Header** and **Footer** (Figure 6-40) and click **OK**.

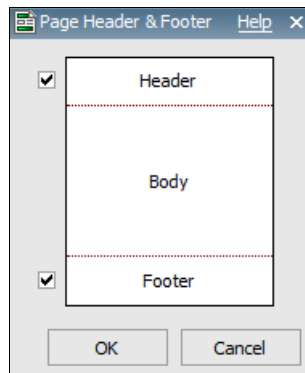


Figure 6-40 Page Header & Footer

5. From the **Insertable Objects** pane, click the **Toolbox** tab.



6. Next, drag a Table object to the Footer area and change the following:
 - Number of columns to 3
 - Number of rows to 1
7. Click **OK** when done.
8. Drag a Text Item object to the left column of the footer table, and in the text dialog box that opens, enter **Report Name** and click **OK** when done.
9. Select the **Report Name** text Item, and in the Properties pane, change the Source Type of the text item to Report Expression instead of text.
10. Click the ellipsis (...) button of the Report Expression property to display the Report Expression dialog box.

- From the available components, click the **Functions** tab and drag the **ReportName** function under the Report Functions folder to the Expression Definition dialog (Figure 6-41), and click **OK** when done.

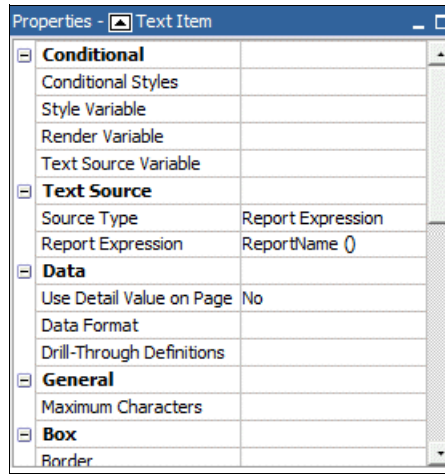


Figure 6-41 Properties pane

- Repeat steps 11 and 12 to add the time and page number to the report footer columns using the PageNumber() and Now() functions

Your footer should look like Figure 6-42 when you are done.

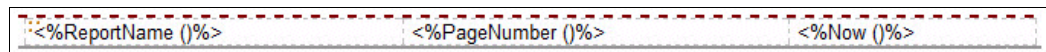


Figure 6-42 Report footer

Now create the header for your report template:

- From the Insertable Objects pane, ensure that the **Toolbox** tab is selected.
- Drag a Table object to the Header area, and change the following:
 - Number of columns to 1
 - Number of rows to 2
- Click **OK** when done.
- Drag an image object to the upper row, select the entire table cell, and click the center button in the toolbar.
- In the Properties pane, under URL Source, click the ellipsis (...) button beside the URL property of the image object, browse for the path of the company logo, and click **OK**.

Important: To browse images on a web server, you must enable Web-based Distributed Authoring and Versioning (WebDAV) on your web server. For more information about configuring web servers, see:

<http://publib.boulder.ibm.com/infocenter/cbi/v10r1m1/index.jsp>

- Drag a **Text Item** to the lower row of the header table.
- In the text dialog, enter Report Title, and click **OK**.
- Select the entire table cell. Then, from the toolbar, click the center button, change the font size to 20 pt, make the text Bold, and change the foreground color to Blue.

9. Click **File** → **Convert to Template**.

Conversion note: The option to convert to template removes any query-related information from the original report, such as data items, filters, and query calculations.

10. From the toolbar, click **Save**, and title the report as `Company Report template` for future use.
11. From the toolbar, click **Run** (Figure 6-43).



Figure 6-43 Report template

Tip: You can add a custom template to be shown with the predefined report templates when creating a new report. For more information, refer to the “Adding Custom Report Templates for Report Studio” section at:

http://publib.boulder.ibm.com/infocenter/cbi/v10r1m1/topic/com.ibm.swg.ba.cognos.ug_cra.10.1.1.doc/c_asg_adding_rstemplates.html

Creating your report

Your company report will consist of the following pages:

- ▶ A cover page that defines the report contents and its use
- ▶ The Map report page that displays the total approved warranty claims per countries on a world map
- ▶ The Charts report page that displays a trend Line chart for the total approved warranty claims by time, and another Pareto chart that displays the number of the total approved warranty claims by product line
- ▶ The List report page that lists the top 10 approved claims by value and the associated client information

Building your report pages

To create the required report pages using your newly created template:

1. Click **File** → **New** from the Cognos Report Studio workbench, select **Company Report template** from the existing folder, and then click **OK**.
2. Next, click **Page Explorer** from the Explorer Bar; By default, Page1 is created.
3. Click the parent **Report Pages** folder (Figure 6-44 on page 201).

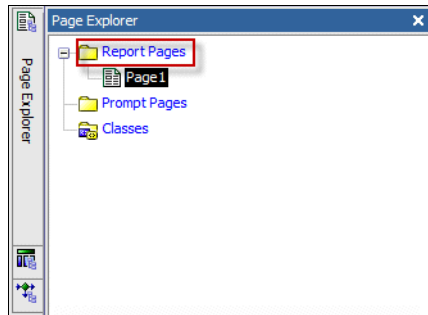


Figure 6-44 Page Explorer

4. Right-click **Page1** in the Report Pages pane, and click **copy**; then, in the white space under Page1, right-click and click **Paste**.

Important: The Page item from the Toolbox adds a new blank page that does *not* follow your template design, so you must copy-paste the design.

5. Repeat step 4 until you have four pages in the Report Pages panel.
6. Click **Page1** and change the Name property from the Properties pane to **Cover**, and repeat this process for the remaining three pages to name each **Map Report**, **Charts Report**, and **List Report**, respectively (Figure 6-45).

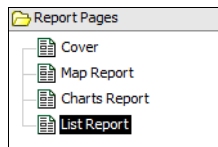


Figure 6-45 Renamed Report Pages

7. Double-click the **Cover** page to open it, and then double-click the **Report Title** text item and enter **Report Cover**.
8. Click **OK** when done,

Create the Map report page

The Map report page displays the total warranty claims per country on a world map.

To build this report page:

1. Click the **Page Explorer** from the Explorer Bar, and then click the **Map Report** page.
2. Double-click the **Report Title** text item and enter **Map Report**.
3. In the Insertable Object pane, ensure that the **Toolbox** tab is selected.
4. Drag a **Table** object to the report work area with number of column = 1 and number of rows = 1.
5. Drag a **Map** object to the table cell.
6. In the Choose Map dialog, confirm that the **World** map is selected with the following options set as outlined here (Figure 6-46 on page 202):
 - Region layers: Countries + Territories
 - Point layers: None
 - Display layers: Oceans

Click **OK**.

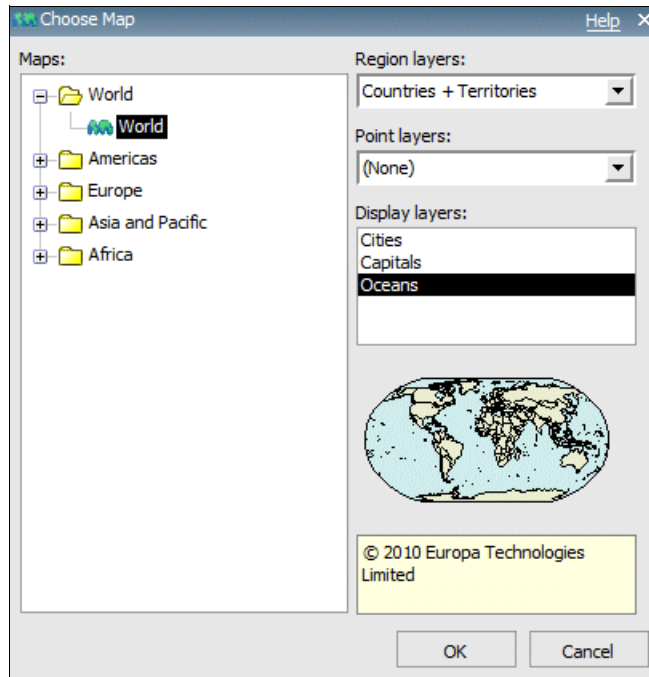


Figure 6-46 Map selection

7. Click the table cell, and select the center button from the toolbar.
8. From the Explorer Bar, click **Query Explorer** and select **Query1**, which enables you to change the query Name to TotalApprovedClaimsPerCountry in the Properties Pane.
9. From the Insertable Objects pane, ensure that the **Toolbox** tab is selected, and drag a **Query Calculation** object to the Data Items pane of the query; enter Claim Requests in the Name of the query calculation object.

10. In the Data Item Expression, enter `count([Publish view].[Warranty Transactions].[WARRANTY_ID])` (Figure 6-47), and then click **Validate** followed by **OK**.

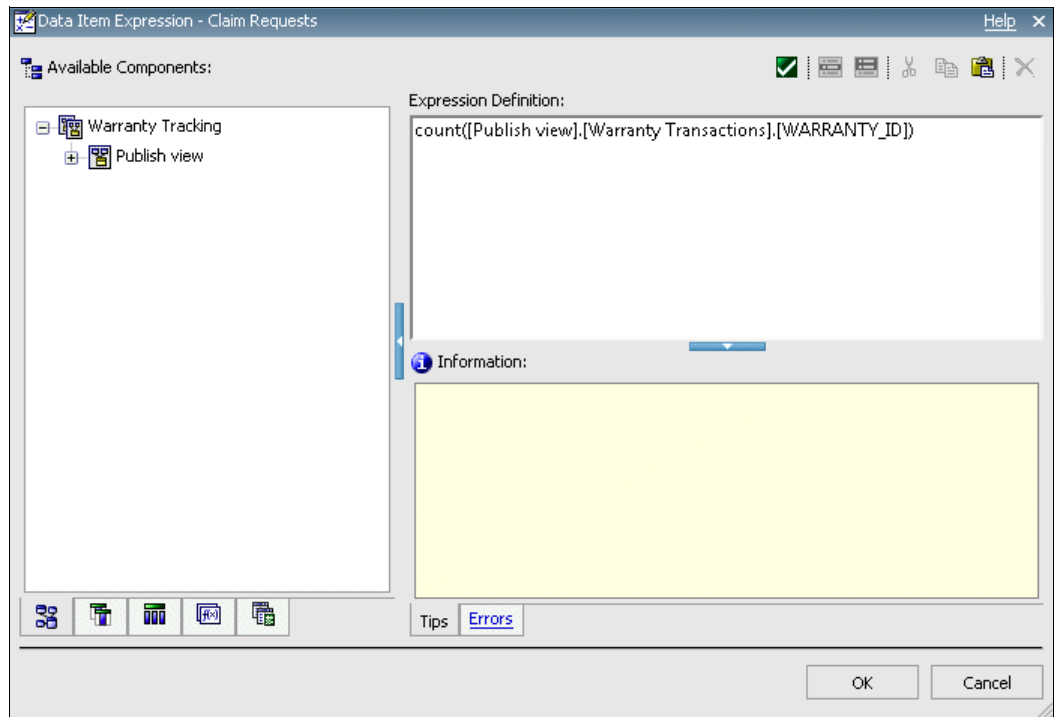


Figure 6-47 Claim Requests Expression

11. From the Insertable Objects pane, ensure that the **Source** tab is selected, and then drag the STATUS_ID data item from Claim Status query subject to the Detail Filters pane (Figure 6-48).

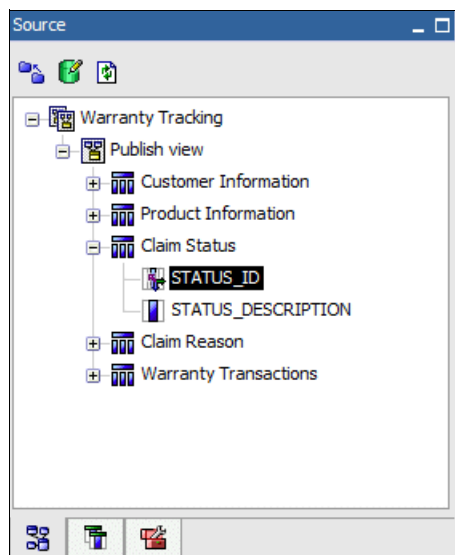


Figure 6-48 Insertable Objects Pane: Source tab

12. In the Detail Filter Expression editor enter the following expression and then click **OK** when done:
- `[Publish view].[Claim Status].[STATUS_ID] = 1`

13. From the Insertable Objects Pane, drag the CUSTOMER_GEOGRAPHY data item to the Data items pane of the query.
14. From the Explorer toolbar, click **Page Explorer** and then click **Map Report**.
15. In the Insertable Objects pane, ensure that the **Data Items** tab is selected (Figure 6-49).

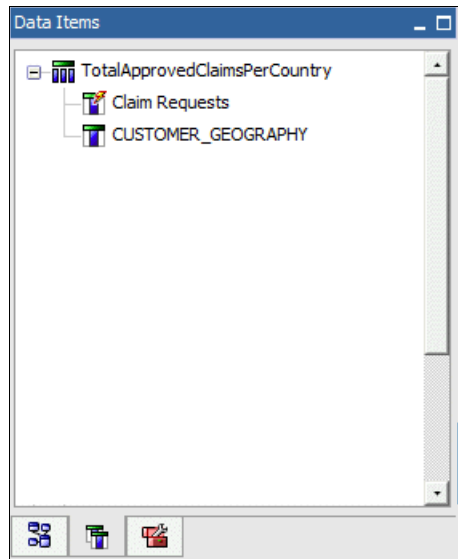


Figure 6-49 Insertable Objects Pane: Data items tab

16. Drag the claim requests data item from TotalApprovedClaimsPerCountry query to the Color drop zone, and then drag the CUSTOMER_GEOGRAPHY data item from the TotalClaimsPerCountry query to the Location drop zone.
17. Click in any white space in the map object to select the entire map report, and then in the Properties Pane change the value of the Title property to *Show* from *Hide*.
18. Change the value of the Name property to *Total approved claims per country (world map)*.
19. In the Work Area, double-click the title of the map text item, enter Total approved warranty claims per country (Figure 6-50) and click **OK** when done.

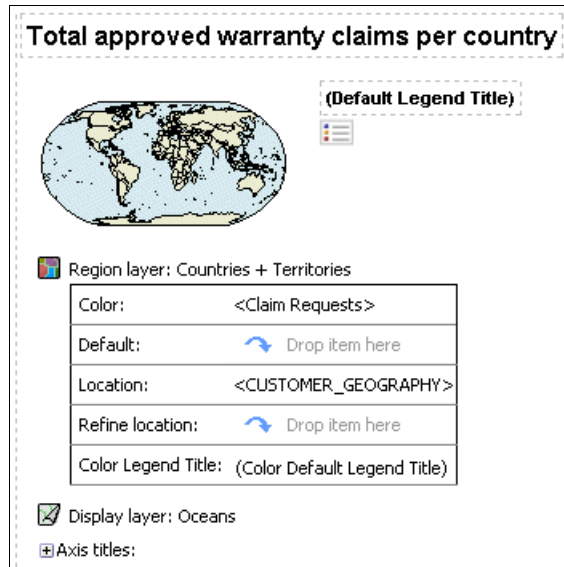


Figure 6-50 Map report

20. In the Map, click **Region layer: Countries and Territories**, and then click the ellipsis button (...) in the Properties pane to open the Palette dialog where you will click **New → Color**, and then adjust the percentage values and colors to be as shown in Figure 6-51.

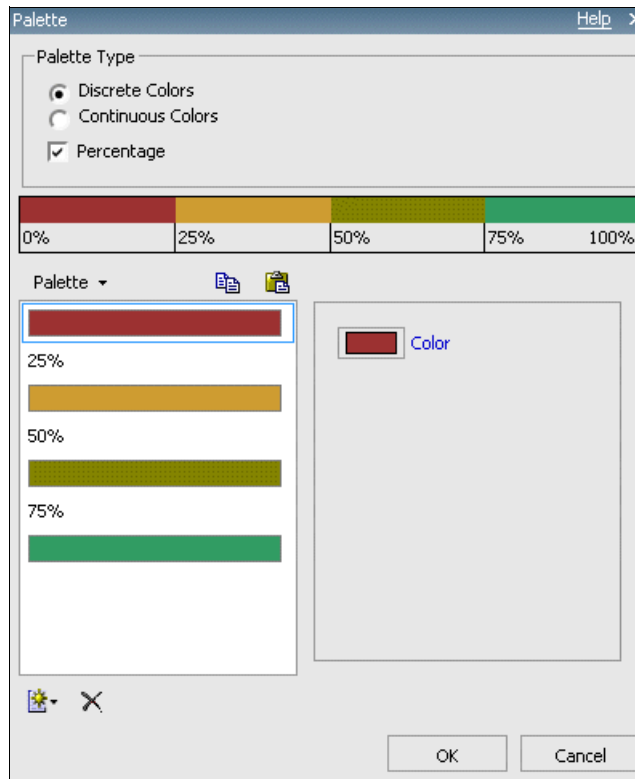


Figure 6-51 Palette colors and percentages

21. Click **OK** when done.

22. From the Toolbar, click **Run** followed by **Page down** until your Map Report page appears as shown in Figure 6-52.

As you can see, the Map report provides a visual summary of the warranty status for each country along with total claims in process. To view the claims totals by country, all you need to do is hover your cursor over the country of interest.

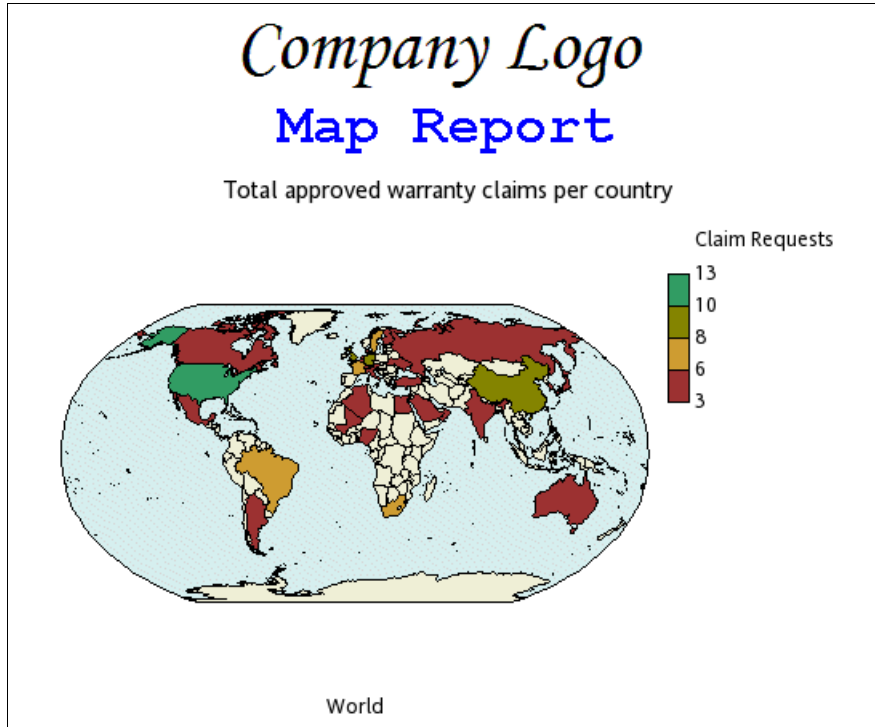


Figure 6-52 Map Report page

Building the Charts report page

As stated above, the Charts report generates two charts: a line and a pareto. The line chart shows the total approved warranty claims by time, while the pareto chart summarizes the number of the total approved warranty claims by product line. This report enables you to understand how long claims take to process and what the frequency of claims for your products is.

Start by building the line chart as follows:

1. From the Explorer Bar, click **Page Explorer** followed by the **Charts Report** page.
2. Double-click the **Report Title** text item and change the text to read: Charts Report.
3. From the Insertable Objects pane, ensure that the **Toolbox** is selected and then drag a **Table** object to the work area, keeping the default settings; click **OK** when done.

- From the Toolbox, drag a **Chart** object to the left column of the table. In the Insert Chart dialog box, select **Line** (Figure 6-53) and click **OK** when done.

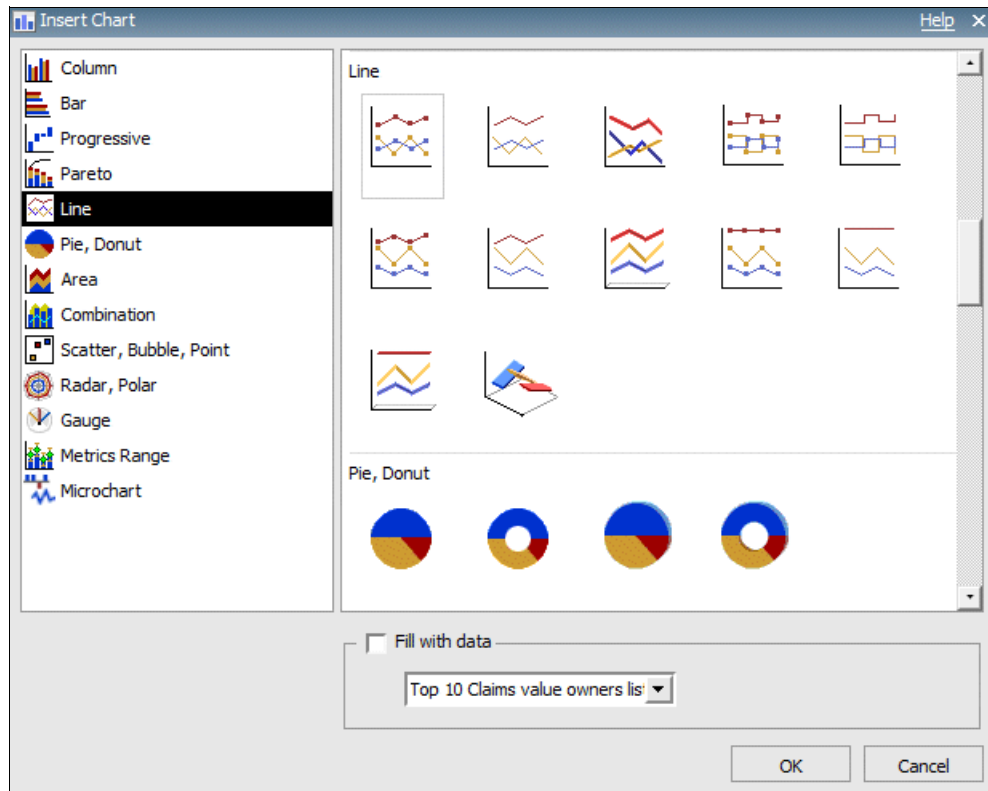


Figure 6-53 Line chart selection

- First, click in any white space inside the Line Chart to select the entire chart, and then change the Title property to *Show* instead on *Hide* from the Properties pane, Chart Titles properties.
- Double-click the **Title** text item in the chart and type Total Approved Claims per month.
- From the Explorer Bar, click the **Query Explorer** page and then click **Query1**. In the Properties pane, change the name to TotalApprovedClaimsForTime.
- From the Insertable Objects pane, ensure that the **Source** tab is selected, expand the **Product Information** query subject and drag the PRODUCT_LINE data item to the Data items pane of the query.
- From the Warranty Transactions query subject, drag the WARRANTY_DATE_TIME_OCCURENCE data item to the Data items pane as well.
- From the Insertable Objects pane, ensure that the **Toolbox** tab is selected, and drag a Query Calculation item to the Data Items pane of the query.
- In the Create Calculation dialog box, type Claim Requests as the Name of the data item.
- In the Data Item Expression dialog box, enter the following expression to count the warranty claims in the Expression Definition, click **Validate** followed by **OK**:
`count([Publish view].[Warranty Transactions].[WARRANTY_ID])`
- Drag a Query Calculation to the Data Items pane of your query, type the name MONTH, and in the expression definition, type the following function to get the month from your time stamp data item, and then click **Validate** followed by **OK** when done:
`month([Publish view].[Warranty Transactions].[WARRANTY_DATE_TIME_OCCURENCE])`

14. Repeat step 13 to add the following expression definition to your Query Calculation to extract the year from your time stamp data item:

```
month([Publish view].[Warranty Transactions].[WARRANTY_DATE_TIME_OCCURENCE])
```

15. Now add the Query Calculation for CHAR_MONTH, which will assign a three-letter abbreviation for each month (Example 6-1).

Example 6-1 CHAR_MONTH Expression

```
CASE WHEN [MONTH] = 1 THEN 'Jan'
      WHEN [MONTH] = 2 THEN 'Feb'
      WHEN [MONTH] = 3 THEN 'Mar'
      WHEN [MONTH] = 4 THEN 'Apr'
      WHEN [MONTH] = 5 THEN 'May'
      WHEN [MONTH] = 6 THEN 'Jun'
      WHEN [MONTH] = 7 THEN 'Jul'
      WHEN [MONTH] = 8 THEN 'Aug'
      WHEN [MONTH] = 9 THEN 'Sep'
      WHEN [MONTH] = 10 THEN 'Oct'
      WHEN [MONTH] = 11 THEN 'Nov'
      ELSE 'Dec' END
```

16. Add a Query Calculation for DATE using the following expression:

```
concat(concat(trim([CHAR_MONTH]),' '),trim(char([YEAR])))
```

17. Select the **Source** tab and then drag the **STATUS_ID** data item from the Claim Reason query subject to the Detail Filters pane.

18. In the Detail Filter Expression editor, type the following expression and click **OK** when done:

```
[Publish view].[Claim Status].[STATUS_ID] = 1
```

Your final query should look like Figure 6-54.

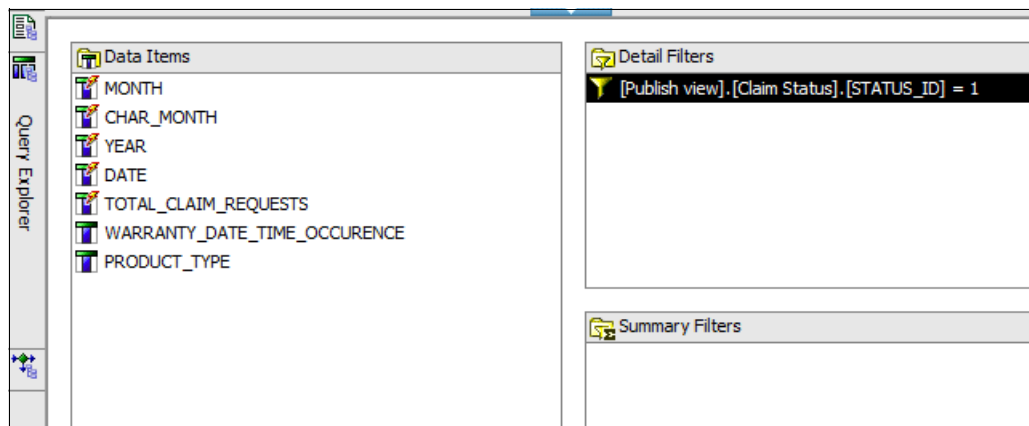


Figure 6-54 TotalApprovedClaimsForTime query

19. From the Explorer Bar, click **Page Explorer** and then select the **Charts Report** page followed by the **Data items** tab in the Insertable Objects pane.

20. From the TotalApprovedClaimsForTime query, drag the following data items to the referenced drop zones:

- **Claim Requests** → **default measure (y-axis)**
- **DATE** → **Categories (x-axis)**

– PRODUCT_LINE → Series

21. Click the **DATE** text on the Categories drop zone until the Properties pane shows **Chart Node Member** beside the select ancestor button, as shown in Figure 6-55.

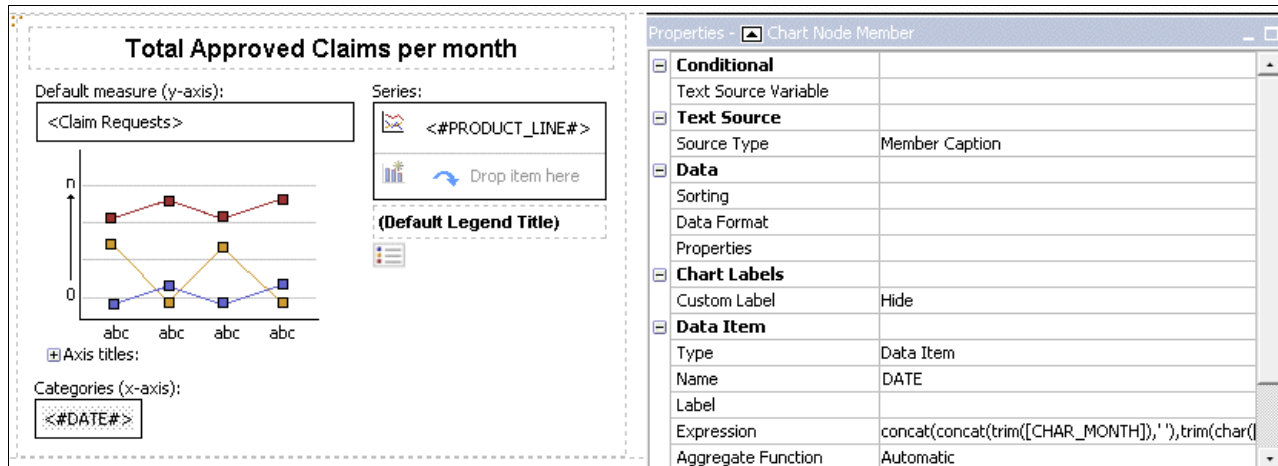


Figure 6-55 Date properties

22. Under the Data section, click the ellipsis (...) button for the Sorting property.

23. In the Sort List, drag the WARRANTY_DATE_TIME_OCCURENCE data item, adjusting the sort order to Ascending (Figure 6-56), and then click **OK** when done.

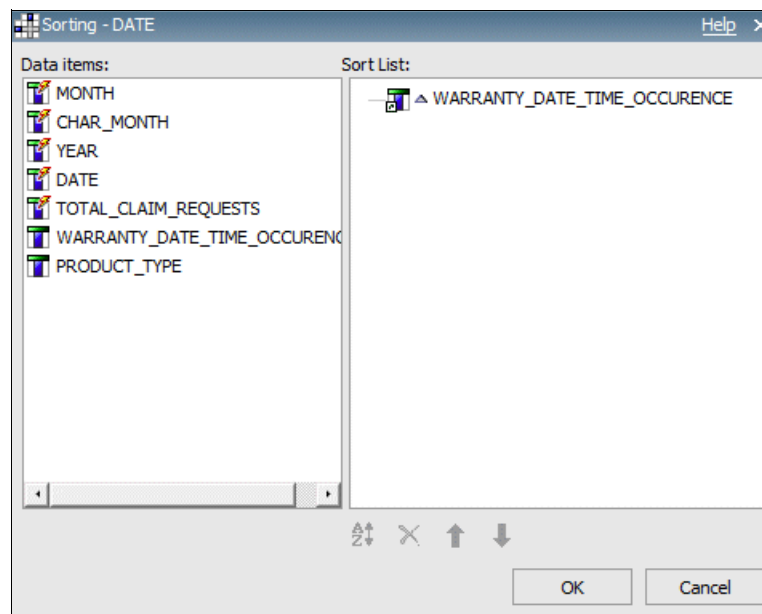


Figure 6-56 Sort List

24. Click in any space on the chart to open the properties pane and change the Name to Total Approved Claims per month Line Chart instead of Combination Chart.

Figure 6-57 on page 210 shows the final design of the chart.

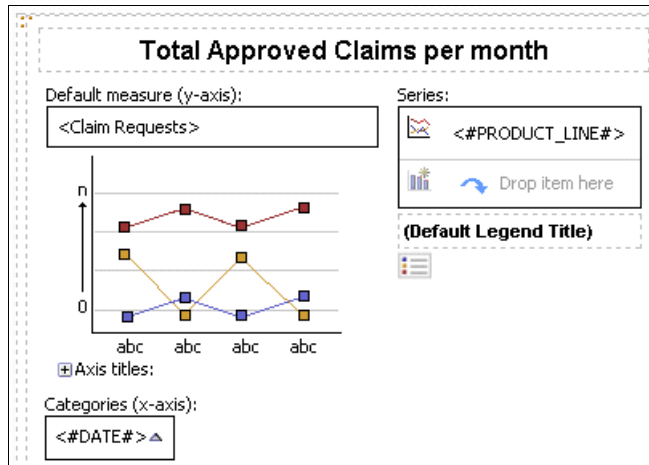


Figure 6-57 Line chart

Now create the Pareto Chart. The reason we opted for it is to plot the factors influencing all warranty claims to better understand this relationship.

1. Select the **Toolbox** tab in the Insertable Objects pane, drag a Chart object to the right column of the table in the work area, choose **Pareto** (Figure 6-58) and click **OK** when done.

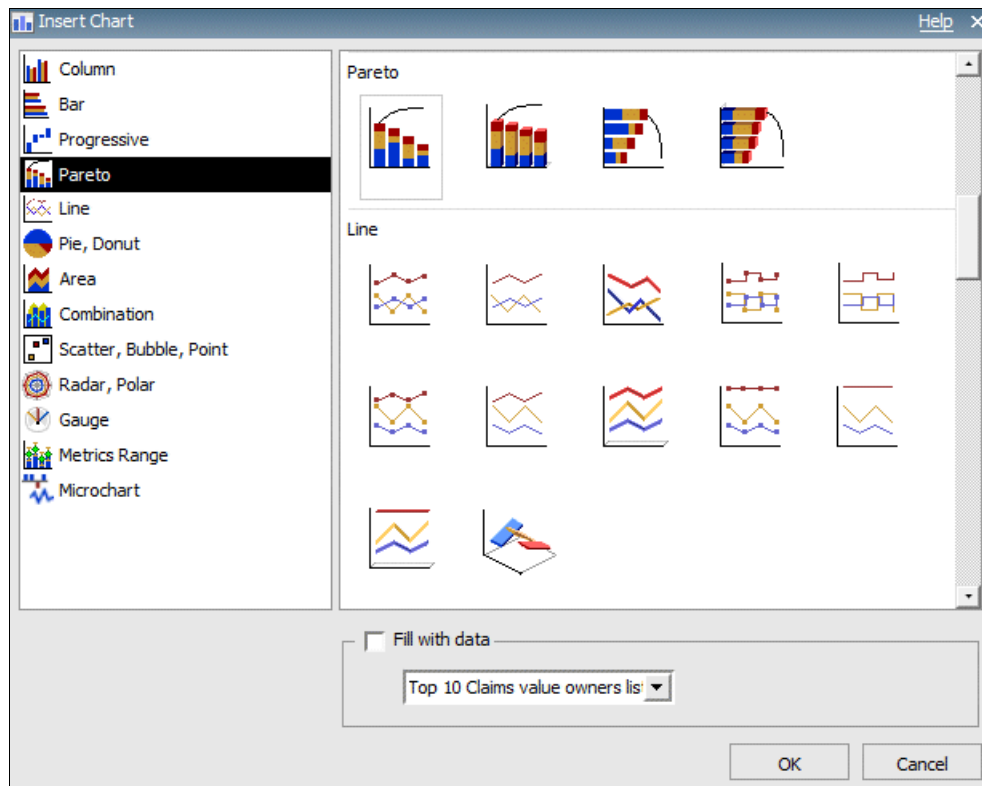


Figure 6-58 Pareto chart selection

2. Click any white space in the Pareto Chart and from the Properties pane under the Chart Titles caption and change the Title property to Show instead of Hide.
3. Double-click the title text item and type Total Approved Claims per product line.

4. From the Explorer Bar, click the **Query Explorer** page followed by **Query1**.
5. In the Properties Pane, change the Name to TotalApprovedClaimsForProductLine.
6. Select the **Source** tab in the Insertable Objects Pane, expand the Product Information query subject and drag the PRODUCT_LINE data item to the **Data items** pane of the query.
7. Expand the Claim Reason query subject and then drag REASON_DESCRIPTION data item to the **Data items** pane of the query.
8. Drag the STATUS_ID data item from the Claim Reason query subject to the **Detail Filters** pane.
9. In the Detail Filter Expression editor, type the following expression and then click **OK** when done:


```
[Publish view].[Claim Status].[STATUS_ID] = 1
```
10. Select the **Toolbox** tab in the Insertable Objects pane and then drag **Query Calculation** to the Data items pane of the query, and type Claim Requests as the Name.
11. Type the following expression in the Data Item Expression dialog box and click **Validate** followed by **OK** when done:


```
count([Publish view].[Warranty Transactions].[WARRANTY_ID])
```

 in the Expression Definition

Your final query should look similar to Figure 6-59.

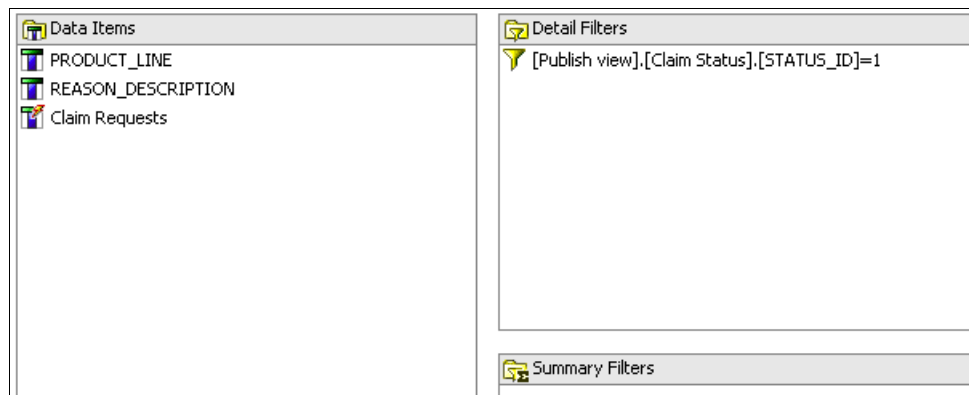


Figure 6-59 TotalApprovedClaimsForProductLine query

12. From the Explorer Bar, click **Page Explorer** and then select the **Charts Report** page followed by the **Data Items** tab in the Insertable Object Pane.
13. From TotalApprovedClaimsForProductLine query, drag the following data items to the referenced drop zones:
 - PRODUCT_LINE → categories (x-axis)
 - REASON_DESCRIPTION → Series
 - TOTAL_CLAIM_REQUESTS → Default measure (y-axis)
14. Click in any white space in the chart to open the properties pane and change the Name to Total Approved Claims per product line (Pareto Chart) instead of Pareto Chart.

Your final chart should look similar to Figure 6-60 on page 212.

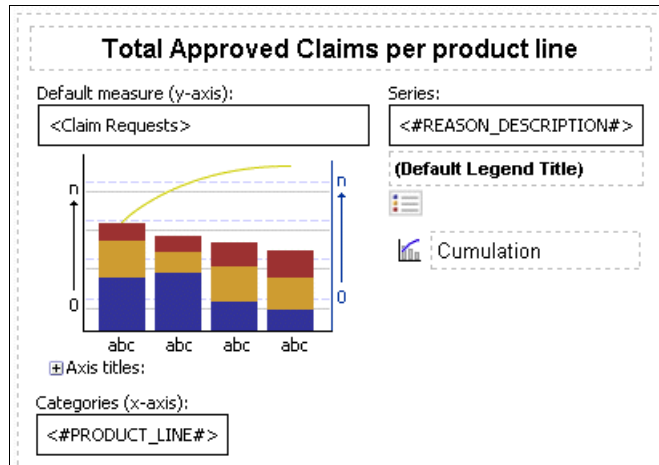


Figure 6-60 Pareto Chart query

15. Next, click the Cumulation icon on the graph and in the Properties pane, click the ellipsis (...) button for the Line Styles property.

16. Adjust the following properties (Figure 6-61):

- Line style to Dash
- Line weight to Point size 2 pt
- Line Color to Black

Click **OK**.

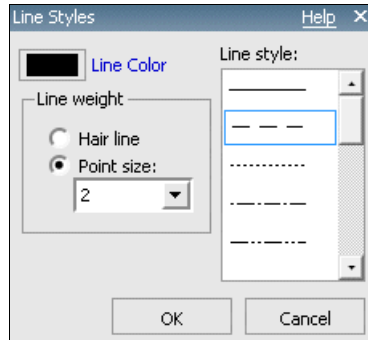


Figure 6-61 Line style

17. From the toolbar, click **Run**, and then click **Page down** until you see the **Charts Report** page (Figure 6-62 on page 213).

The Line Chart here indicates that June and July are the highest months for claim requests. Armed with this information, a business user might investigate the cause for these claims to determine whether the products are faulty and/or there is an issue in the business process. The Pareto chart will help the business user validate the causation quickly, as it provides a concise summary of the common reasons for claim submissions by product.

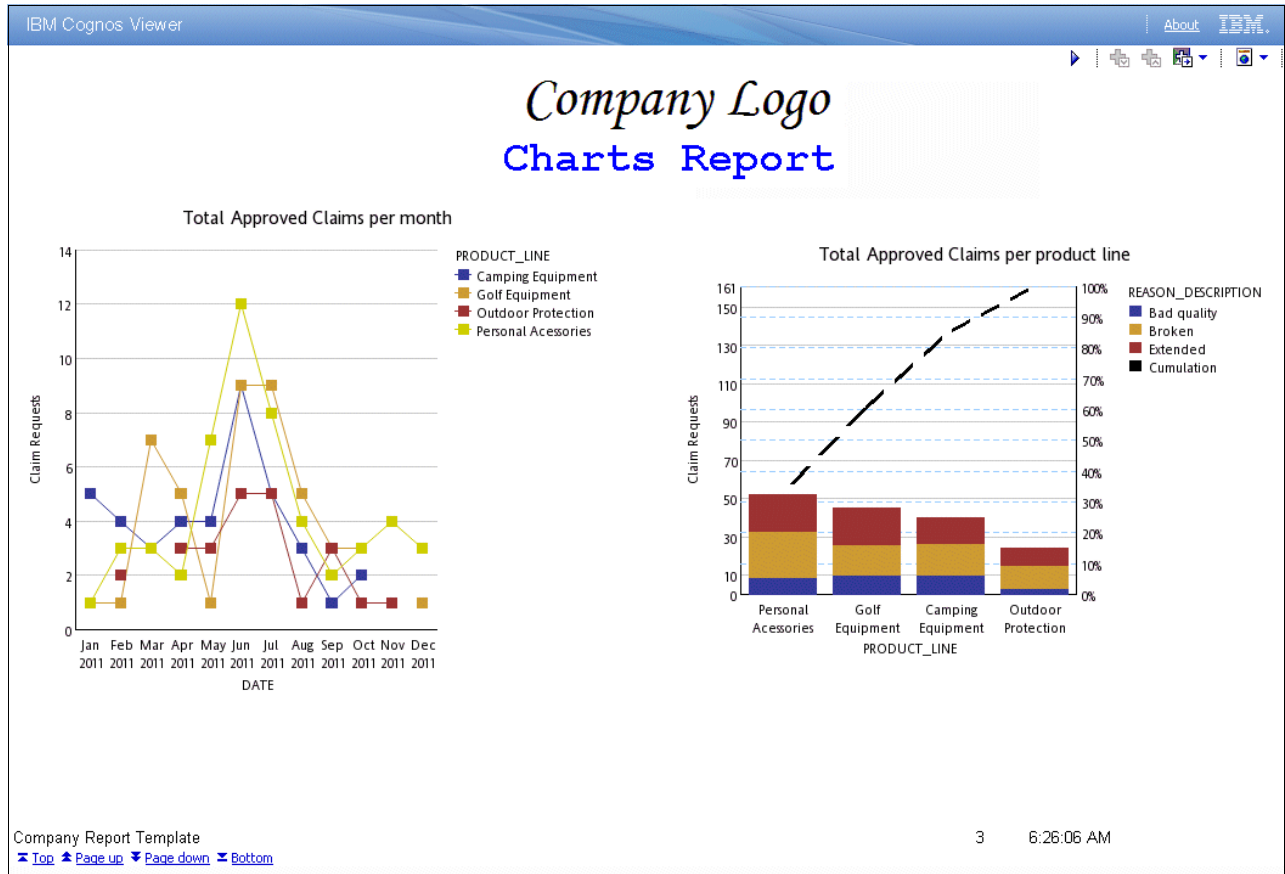


Figure 6-62 Charts Report page

Create the List report page

This report displays the top 10 claims from the highest product value along with client information. To create this report:

1. From the Explorer Bar, click **Page Explorer** followed by the **List Report** page, and then double-click **Report Title** and type List Report.
2. Select the **Toolbox** tab in the Insertable Objects pane, drag a Table object to the work area and change the number of columns to 1 and the number of rows to 1; click **OK** when done.
3. Drag a Block object to the table in your work area.
4. Drag a Text Item object inside the block, and type Top 10 Approved Warranty Claims Value Owners in the text dialog box.
5. Select the entire block, and then change the following information from your Toolbar:
 - The font to Bold
 - Font size to 14 pt
 Then click the center button.
6. Drag a new Block object below your current object, and then drag a List object inside it.
7. Click the center button from the Toolbar, and then select the entire list from the **Properties** pane and change the Name of the List to Top 10 Approved Claims value owners list.

8. Select the **Source** tab from the Insertable Objects pane, and drag the following data items from the referenced query subjects, as shown in Figure 6-63:
 - WARRANTY_ID → Warranty Transactions to the first column of the List.
 - CUSTOMER_ID → Customer Information to the next column of the List.
 - PRODUCT_VALUE → Product Information to the next column of the List.



Figure 6-63 List Report

9. Select the PRODUCT_VALUE list header followed by Sort from the toolbar and select **Descending** order (Figure 6-64).

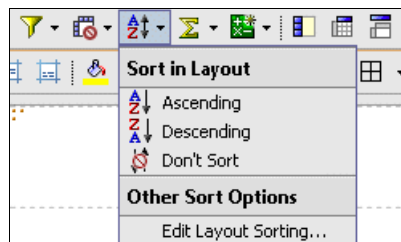


Figure 6-64 Sorting menu

10. Select any column in the list to open the **Properties** pane and click the **Select Ancestor** button and choose **List**.
11. From the Explorer toolbar, click the **Query Explorer** tab and then click **Query1** to open the **Properties** pane and change the name of the Query to Top_10_ApprovedClaimsValueOwners.
12. Select the **Source** tab in the Insertable Objects pane and then drag the STATUS_ID data item from the Claim Status query subject to the Detail Filters pane.
13. In the Detail Filter Expression editor type, type the following expression and click **OK** when done:


```
[Publish view].[Claim Status].[STATUS_ID] = 1
```
14. Select the **Toolbox** tab in the Insertable Objects pane, drag a Query Calculation item to the Data items pane of the query and change its name to Rank.
15. In the Expression Definition type the following expression and click **Validate** followed by **OK** when done:


```
rank([PRODUCT_VALUE])
```

16. Drag the Rank item to the Detail Filters pane and enter [Rank] <= 10 in the Expression Definition as shown in Figure 6-65 to return the first top 10 ranked customers who requested claims by the product value.

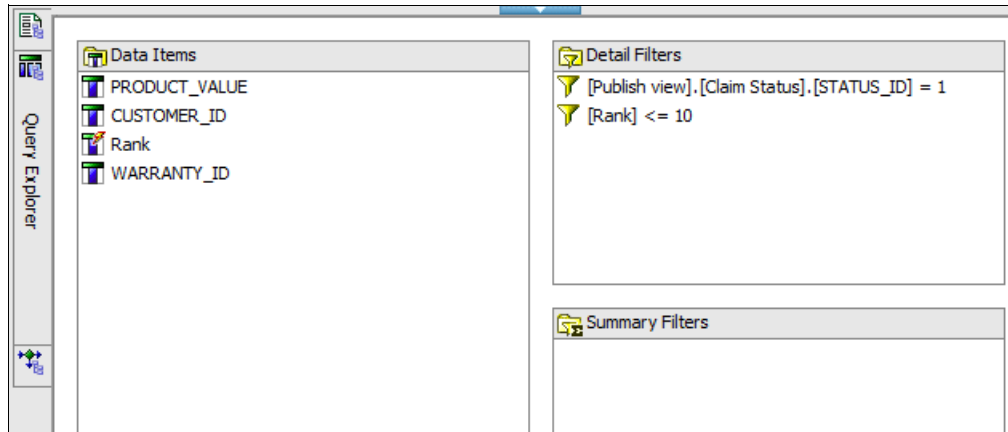


Figure 6-65 Top_10_ApprovedClaimsValueOwners query

17. From the toolbar, click **Run** and then page down until your List Report page displays (Figure 6-66).

Company Logo
List Report
Top 10 Approved Warranty Claims Value Owners

WARRANTY_ID	CUSTOMER_ID	PRODUCT_VALUE
171	4221	476
172	4085	476
175	4223	476
176	4087	476
276	4330	476
279	4334	476
326	4492	476
327	4482	476
183	4268	473.07
184	4091	473.07
187	4178	473.07
217	4175	473.07
218	4121	473.07
322	4493	473.07
323	4483	473.07

Figure 6-66 List Report page

Notice that more than 10 rows are retrieved. The first eight rows have the same rank of 1 because all have the same product value. The other seven rows have the same rank of 9

because they too have the same product value. By default, the Cognos is programmed to present all data of similar rank instead of trimming a list to only 10.

18. From the File menu, click **Save** to save your report as The Company Report.

6.4.1 Creating Active Reports with Cognos Report Studio

You can use IBM Cognos Report Studio to create *active reports*, which is a new reporting style introduced in IBM Cognos 10 Business Intelligence. An active report provides interactive access to data in an easy-to-use format. Users can explore their data and derive additional insight without additional involvement from IT. In addition, active reports can be consumed by users who are offline, making them an ideal solution for remote users such as the sales force.

To make an effective active report, take precautions regarding the amount of retrieved data, which can impact the file size. Consider reducing the maximum number of rows. By default, only 5000 rows are retrieved. The file size is also impacted by the number of charts in the report, because all charts are kept in the file and rendered at runtime. The preferred file size is determined by the target audience access preference (that is, workstation vs. iPad). To create a simple active report that shows the top 10 warranty claim values with their customer information:

1. Click **Launch** from the Cognos Connection and select **Report Studio**.
2. In the Package select dialog box, click **Warranty Tracking** → **Create new report** → **Active Report** (Figure 6-67), followed by **OK**.

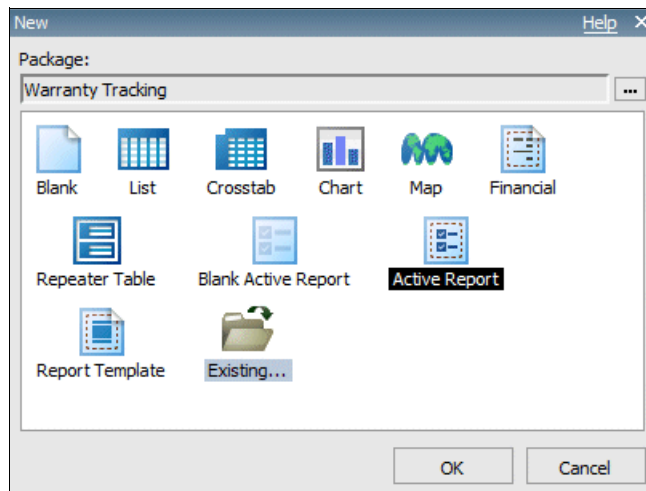


Figure 6-67 Select Active Report

3. Select the **Toolbox** tab in the Insertable Objects pane, and then drag a List to the report area.
4. From the **Source** tab, expand the Warranty Transactions query subject, and add the WARRANTY_ID data item to the list.
5. Then, expand the Customer Information query subject and add the CUSTOMER_ID to the list.
6. Finally, expand the Product Information query subject and add the PRODUCT_VALUE data item to the list.
7. Click the PRODUCT_VALUE column header followed by **Sorting** from the toolbar, and then click **Descending**.

8. From the Explorer bar, select the **Query Explorer** tab and click **Query1**.
9. From the Toolbox tab, drag Query Calculation to the Data Items tab.
10. Enter Rank in the Name of the new query calculation, and in the expression editor window, enter the expression `rank([PRODUCT_VALUE])`.
11. From the Data Items pane of the query, drag the Rank data item to the Detail Filters tab, and enter `[Rank] <= 10` in the expression editor dialog box, because you want to reduce the size of the report so that only the products with a value more than 100 are considered.
12. Drag the PRODUCT_VALUE data item from the query data items to the Detail Filters tab, and then enter `[PRODUCT_VALUE] > 100` and click **OK** when done.
13. Drag the STATUS_ID data item from the Claim Status query subject to the Detail Filters pane.
14. In the Detail Filter Expression editor type, enter the following expression and click **OK** when done:
`[Publish view].[Claim Status].[STATUS_ID] = 1` (Figure 6-68)

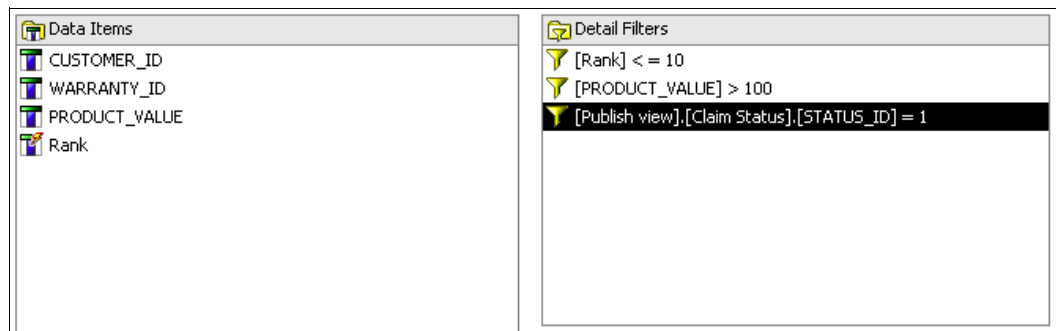


Figure 6-68 Active report query

15. Click **Query1** from the Query Explorer Page, and change the Name property to `Top_10_Claims`.
16. Then, from the Page Explorer, click **Page1** and from the Toolbox tab, drag a block to the left of the list.
17. Right-click in the Toolbox pane, and select **Active Report Toolbox Items**.
18. Drag the Data Button Bar to the block; you can customize the interactive components of the active report to have it display your list according to the product line.
19. From the Source tab, expand the Product Information query subject and then drag `PRODUCT_LINE` to the bar drop zone (Figure 6-69 on page 218).

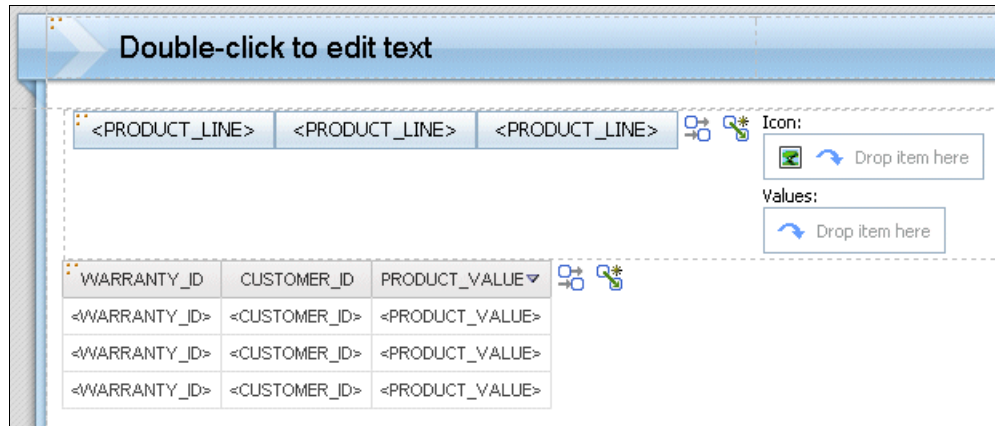


Figure 6-69 Active Report design

20. To create the connection between the data button bar and the list so that when the user clicks a product line button, the list is filtered for this product type only, click the **Interactive Behavior** button beside the data button bar (Figure 6-70).

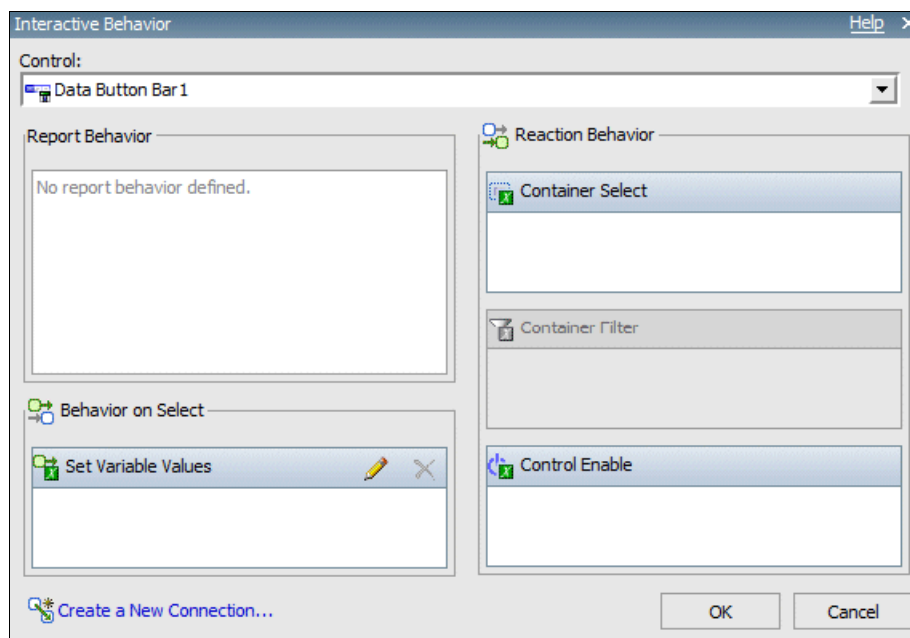


Figure 6-70 Interactive Behavior

21. Click the **Create a New Connection** link to open a dialog box prepopulated with the two objects on the report page. Change the Behavior value to `filter`, and change the variable name to `PRODUCT_LINE_PARAM` under the Active Report variable, and then click **Connect** when done. Refer to Figure 6-71 on page 219.

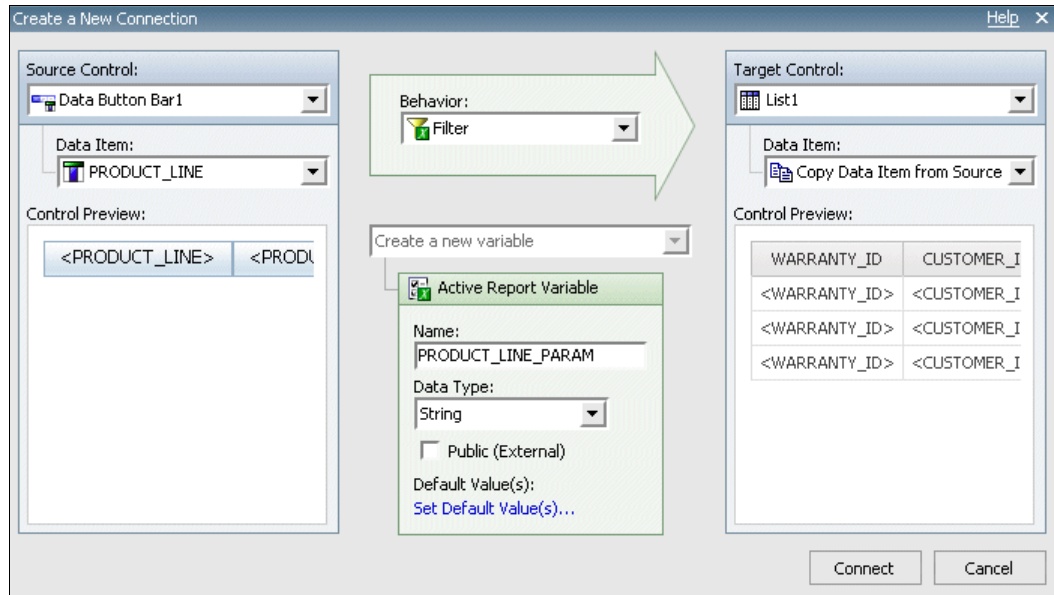


Figure 6-71 Create connection

22. Click **OK** in the Interactive Behavior dialog box, and double-click the text item in the report page header and type Top 10 claims per product line in the text item dialog box (Figure 6-72); click **OK** when done.

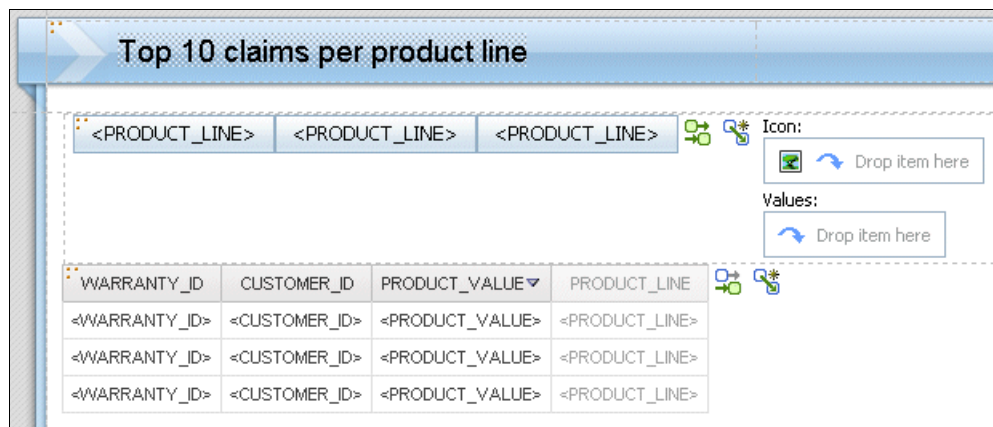


Figure 6-72 Active Report design

23. Click **Run** from the toolbar to view the output as shown in Figure 6-73 on page 220 and Figure 6-74 on page 220.

Top 10 claims per product line				
Camping Equipment		Golf Equipment	Outdoor Protection	Personal Accessories
WARRANTY_ID	CUSTOMER_ID	PRODUCT_VALUE		
172	4085	476		
176	4087	476		
171	4221	476		
175	4223	476		
276	4330	476		
279	4334	476		
327	4482	476		
326	4492	476		
184	4091	473.07		
218	4121	473.07		
217	4175	473.07		
187	4178	473.07		
183	4268	473.07		
323	4483	473.07		
322	4493	473.07		

Figure 6-73 Active report page

Top 10 claims per product line				
Camping Equipment		Golf Equipment	Outdoor Protection	Personal Accessories
WARRANTY_ID	CUSTOMER_ID	PRODUCT_VALUE		
172	4085	476		
176	4087	476		
171	4221	476		
175	4223	476		
276	4330	476		
279	4334	476		
327	4482	476		
326	4492	476		

Figure 6-74 Active report for Personal Accessories product line

24. From the File menu, click **Save** and save the report as “Top 10 approved claims Active Report.”

6.5 Creating a dashboard with Cognos Business Insight

IBM Cognos 10 Business Insight is a web-based tool introduced in the IBM Cognos Business Intelligence 10.1.1 suite. It is designed for business users who want to get insight by building interactive dashboards. It is also integrated with social business tools to facilitate collaborative

decision making. Cognos Business Insight is not a tool to develop new reports, but it can be used to combine information from already rebuilt reports or other sources and combine them into a dashboard interface.

For more information about IBM Cognos 10 Business Insight, see the IBM Business Insight User Guide at:

http://public.dhe.ibm.com/software/data/cognos/documentation/docs/en/10.1.1/ug_buxc.pdf

Based on the scenario in this chapter, our business users want to create a dashboard that can show the current state of the business.

To create this dashboard:

1. Click **Launch** in IBM Cognos Connections followed by **Business Insight** → **Create New** to open a new workspace.
2. Select the **Content** tab in the Content pane and expand My Folders to access the Company Report and Top 10 approved claims Active report.
 - Notice the subcomponent names of the Company Report (Figure 6-75). Cognos Business Insight deals with every object of the report pages as a widget.

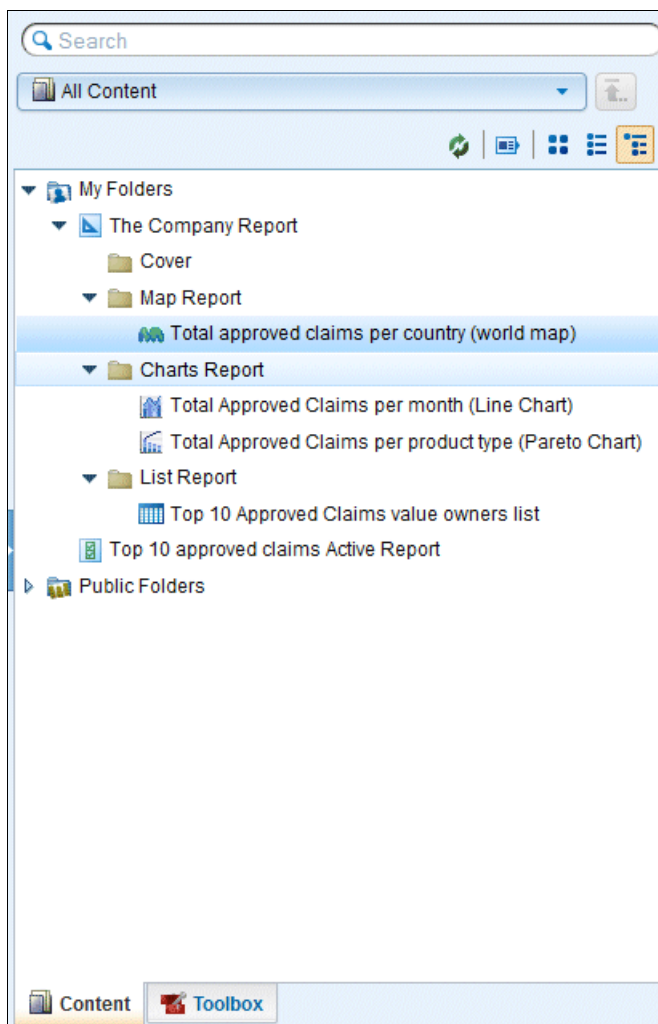


Figure 6-75 Content pane

- From the Content tab, drag both Total Approved Claims per month (Line Chart) and Total Approved Claims per product line (Pareto Chart) to the workspace (Figure 6-76).

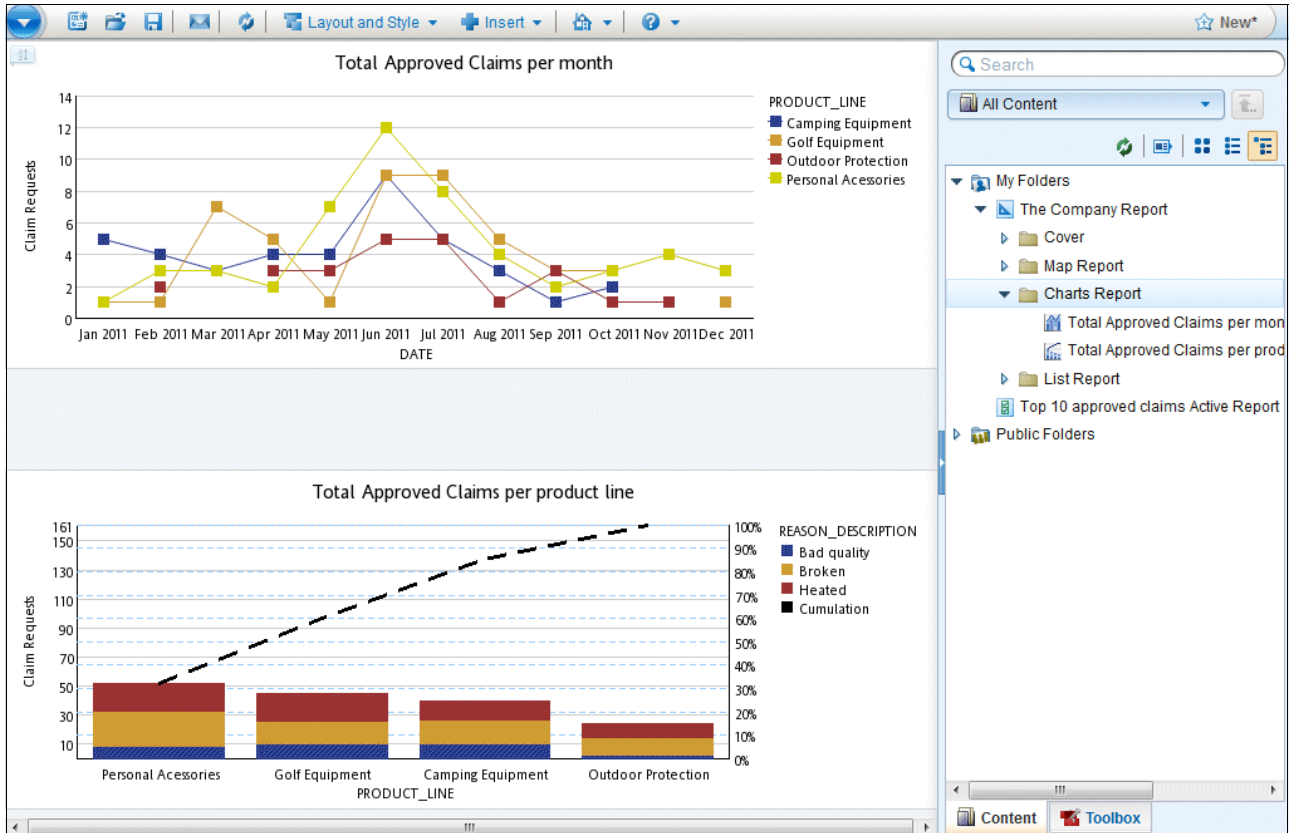


Figure 6-76 Cognos Business Insight workspace

- Select the **Toolbox** tab in the Content pane and drag a Slider Filter to your workspace.

- In the Properties window, select the PRODUCT_LINE data item that is listed in the Data item column of the “Select a data item to filter on” list. The Widget title column lists the corresponding widgets that will be affected by this slider on that data item; keep the remaining defaults (Figure 6-77) and click **OK** to proceed.

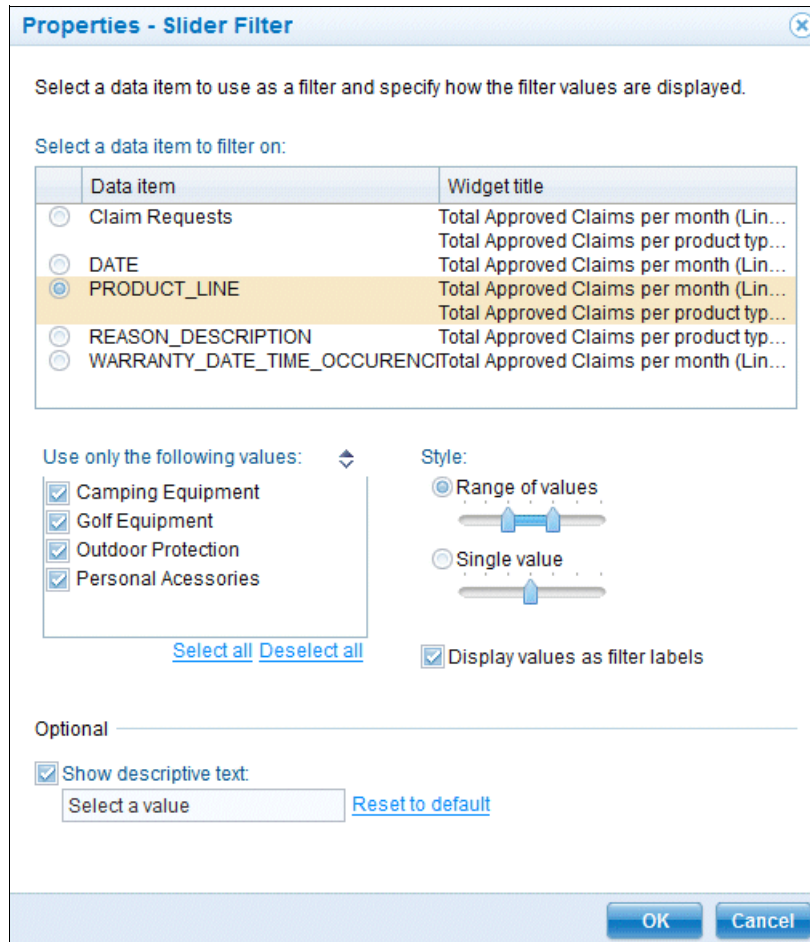


Figure 6-77 Properties window

- Align all the widgets manually or right-click any space beside the widgets and click **Fit All Windows to Window**.
- Select both the Camping Equipment and Golf Equipment product lines from the slider filter, while noticing the interactive communication on the other widgets (Figure 6-78 on page 224), and then click **Save** from the Application bar and name it Dashboard.

Note: The business user with minimal technical background can use IBM Cognos Business Insight to build interactive dashboards and measure the performance of the organization effectively.

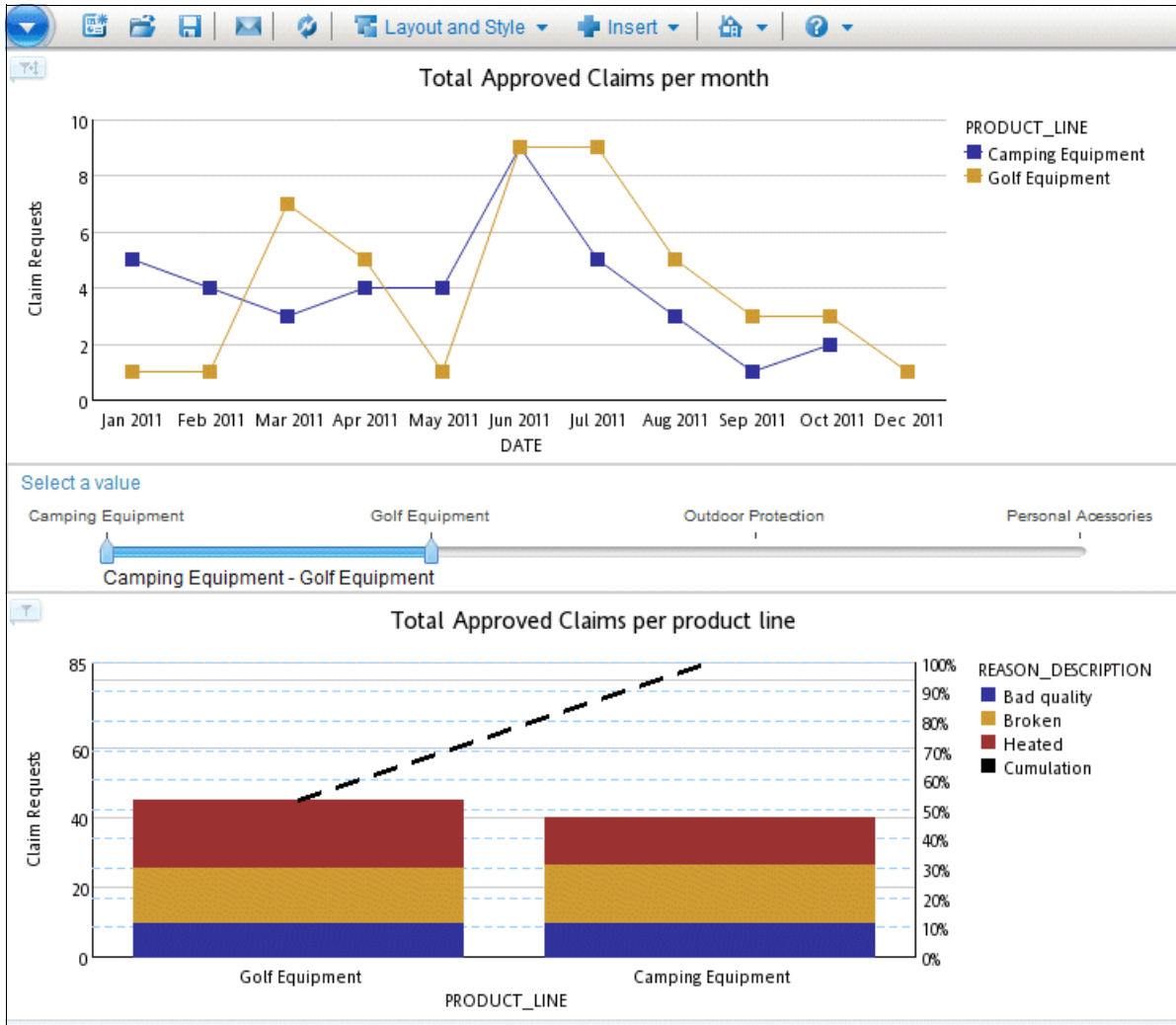


Figure 6-78 Dashboard

6.6 Authoring events with Cognos Event Studio

It is critical for decision makers to be alerted when specific situations are detected. Prompt action can help save business deals. IBM Cognos Event Studio is the notification tool used to alert decision makers in your organization of events as they happen, so that they can make timely and effective decisions.

You can use Event Studio to create agents that monitor status changes, priority customers, your organization's data to detect occurrences of business events, or any other factor that is important to your business (Figure 6-79 on page 225). When an agent detects an event, it can perform tasks, such as sending an email, adding information to the portal, and running reports.

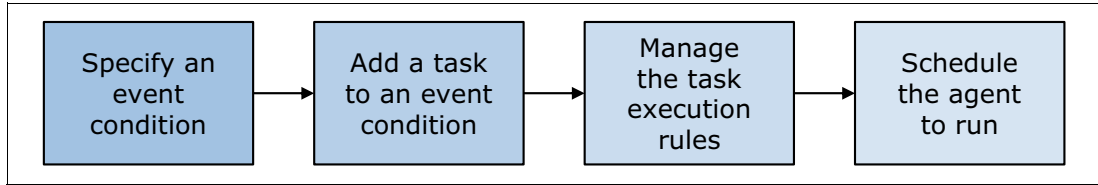


Figure 6-79 Cognos Event Studio process life cycle

For more information about IBM Cognos 10 Event Studio, see the IBM Business Intelligence Cognos 10.1.1 Information Center at:

http://publib.boulder.ibm.com/infocenter/cbi/v10r1m1/topic/com.ibm.swg.ba.cognos.wig_cr.10.1.1.doc/c_gtstd_es.html

In our sample, the warranty executives need periodic updates detailing warranty claims where the associated product values are equal to or higher than 100. The executives want to receive email updates showing those requests with the clients' information.

6.6.1 Specify an event condition

The first step is to define the event condition triggering the event.

1. From the IBM Cognos connection, click **Launch**, and then click **Event Studio**.
2. Choose the package **Warranty Tracking**.

Note: By default, the **I want to** pane refers to the **Specify an event condition state** as shown in Figure 6-80.

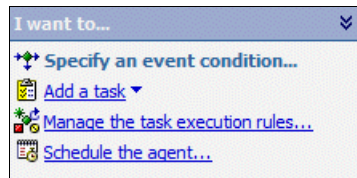


Figure 6-80 I want to list

3. Because WARRANTY_ID is unique over the system, adding the function total for the report value for each WARRANTY_ID reports only the single product value of the associated WARRANTY_ID. Select the **Detail** tab in the Expression pane and type total . From the Insertable Objects pane, select **[PRODUCT_VALUE]** from the Product Information query subject and click **Insert**.
4. Enter for in the Expression pane, select **[WARRANTY_ID]** from the Warranty Transactions query subject and click **Insert**. Enter >= 100 in the Expression pane.

Your final expression should be total ([PRODUCT_VALUE] for [WARRANTY_ID]) >= 100 as shown in Figure 6-81 on page 226.

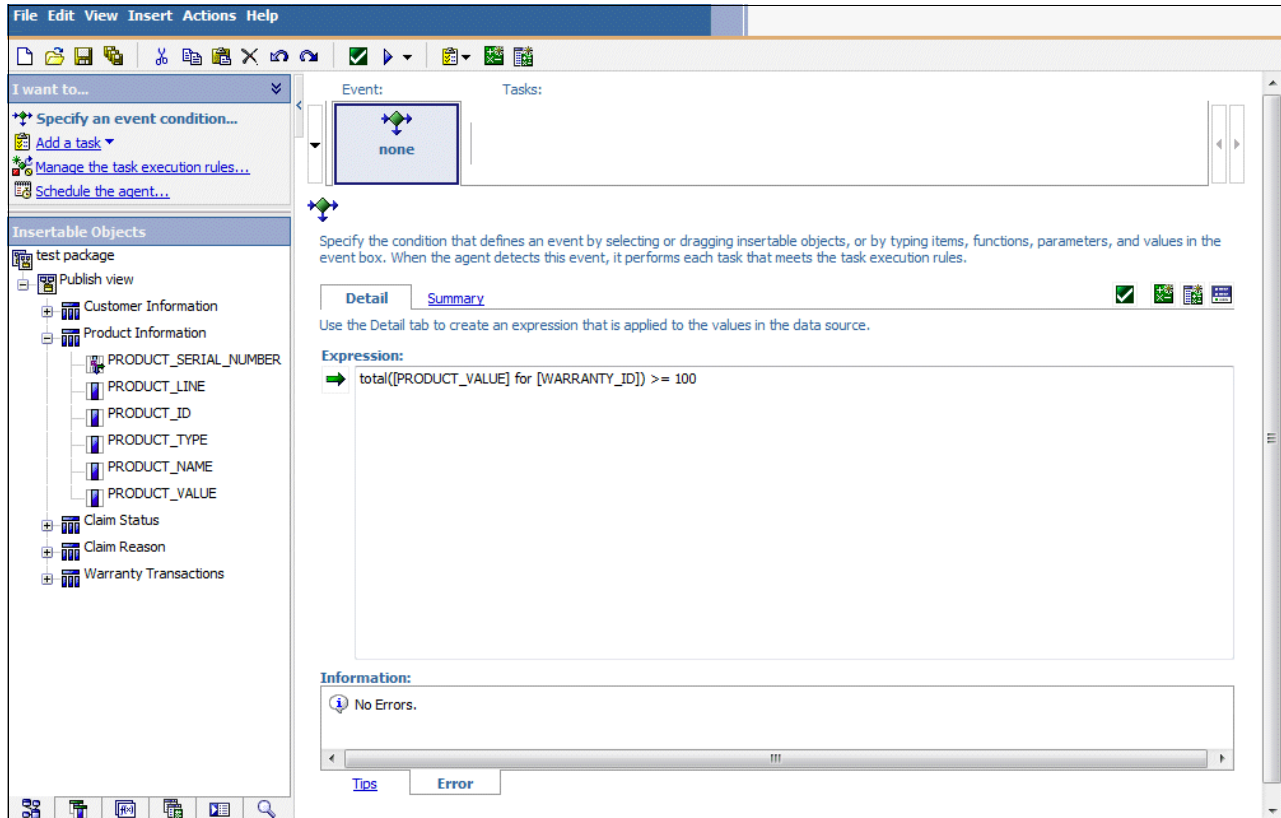


Figure 6-81 Cognos Event Studio window

5. Click **Validate** and then click **Preview** in the toolbar.

You can see the warranty transactions where their product values exceed 100, as shown in Figure 6-82 on page 227.

IBM Cognos Viewer	
PRODUCT_VALUE	WARRANTY_ID
186	158
186	159
476	171
476	172
476	175
476	176
392.57	179
392.57	180
473.07	183
473.07	184
473.07	187
250	189
250	190
250	207
250	208
250	211
473.07	217
473.07	218
250	225
250	226

[Top](#)
[Page up](#)
[Page down](#)
[Bottom](#)

Figure 6-82 Preview event condition

6.6.2 Add a task to an event condition

The second step in this process is to add a task to associate with the event condition, and there needs to be at least one. Some of the tasks are notification tasks, while others are workflow-related tasks. By default, tasks run at the same time in parallel, but you can specify that each task be performed in an order of your choice.

Note: Some of the tasks, such as run an export, run an index update, and run an import, are for administrator use only.

To create a job:

1. From Cognos Connection, click **New Job**.
2. In the Job Wizard, enter Active Report job in the Name field, specify the job location as shown in Figure 6-83 on page 228, and click **Next**.

Specify a name and description - New Job wizard

Specify a name and location for this entry. You can also specify a description and screen tip.

Name:
Active Report job

Description:

Screen tip:

Location:
My Folders
[Select another location...](#)

Cancel < Back Next > Finish

Figure 6-83 Job wizard

3. Click the add link to navigate to Top 10 Claims Active Report, select the report, click **Add** followed by **OK** to proceed.
4. To change the run options, click the set icon.
5. In the select the report options, select the following options and click the **Edit the options** link (Figure 6-84):
 - **Override the default values**
 - **Send a link to the report by email.**

Select the report options - New job - Top 10 approved claims Active Report Help X

Select the run options to use for this step.

Run the report to:
 Produce report outputs ▼

Report options:

Override the default values

Format:
HTML

Accessibility:
 Enable accessibility support

Languages:
 English (United States) [Select the languages...](#)

Delivery:
 Select at least one delivery method. For burst reports, the email recipients are determined by the burst specification.

Save:
 Save the report
 Save the report as a report view [Edit the options...](#)
 ... > Report View of Top 10 approved claims Active Report

Send a link to the report by email [Edit the options...](#)
 0 recipients

Prompt values:
 Override the default values
 No values saved

OK Cancel

Figure 6-84 Select the report options wizard

6. Add the email recipients, Subject, body, select the **Attach the report** box and deselect **Include a link to the report** as shown in Figure 6-85; click **OK** to proceed

To:
[EMAILS]

Cc:
[Select the recipients...](#) [Show Bcc](#)

Subject:
Warranty values exceeds 100 owners

Body: [Change to plain text >>](#)

Dear Executives,
 Attached the updated report shows the top clients have warranty requests which exceeds 100.

Include a link to the report
 Attach the report

OK Cancel

Figure 6-85 Email form

7. Click **OK** in the Select the Run Options wizard followed by **Next** in the Select the Steps wizard, and then click **Save only** followed by **Finish**.

Now, create a task to run your job from Cognos Event Studio:

1. From the **I want to** menu, click **Add a task** → **Run a job**.
2. Then, select the Active Report job (Figure 6-86) and click **OK**.

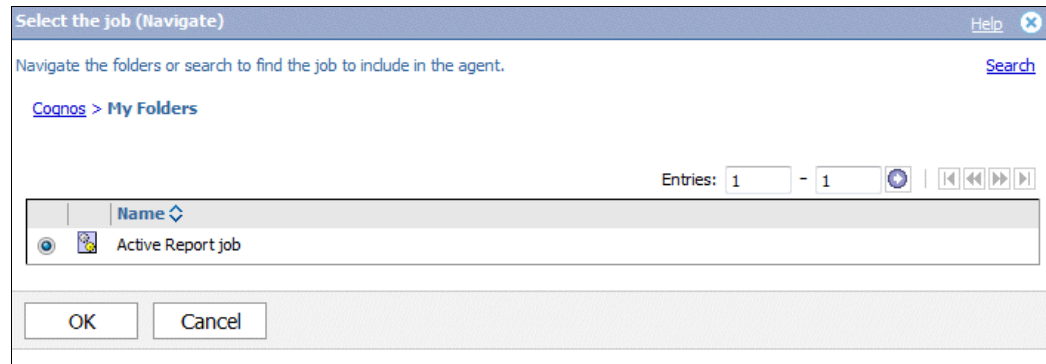
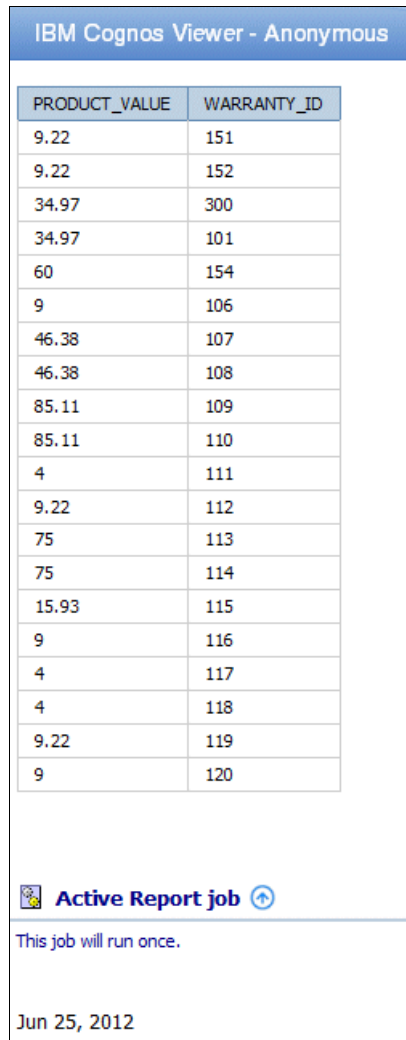




Figure 6-86 Select the job

3. From the toolbar, click **Preview** (Figure 6-87).



IBM Cognos Viewer - Anonymous

PRODUCT_VALUE	WARRANTY_ID
9.22	151
9.22	152
34.97	300
34.97	101
60	154
9	106
46.38	107
46.38	108
85.11	109
85.11	110
4	111
9.22	112
75	113
75	114
15.93	115
9	116
4	117
4	118
9.22	119
9	120

 **Active Report job** 

This job will run once.

Jun 25, 2012

Figure 6-87 Preview task

6.6.3 Manage the task execution rules

At this point, you want to specify which event triggers the agent to run the email job. To manage the task execution rules:

1. From the **I want to** area, select **Manage the task execution rules**.
2. In the Event Selection tab, by default the tasks run for New and Ongoing events for the selected event, so select only **New events** and clear the Ongoing events (Figure 6-88 on page 232).

Event Selection

Event Key

Select a task and specify when the agent performs it. You can specify to perform the task for selected events, or when the agent or its tasks fails.

Task:
Active Report job (job) ▼

Select when to perform this task:

For selected events:

✱ New events

↻ Ongoing events:
Specify whether the task is performed for all ongoing events or only when the values of selected items change or do not change.

All ongoing events ▼

➔

[Remove](#)

■ Ceased events

⚡ When the following item is True for selected new, ongoing, and ceased events
Insert an item or calculation that evaluates to True or False.

➔

[Remove](#)

⚡ No events

❌ When the agent or any of its tasks fail

OK

Cancel

Figure 6-88 Event selection

3. In the event condition, WARRANTY_ID is unique; choose it as the event key.
4. In the Event Key tab, click **Include only selected items**, remove PRODUCT_VALUE as shown in Figure 6-89 on page 233, and then click **OK**.

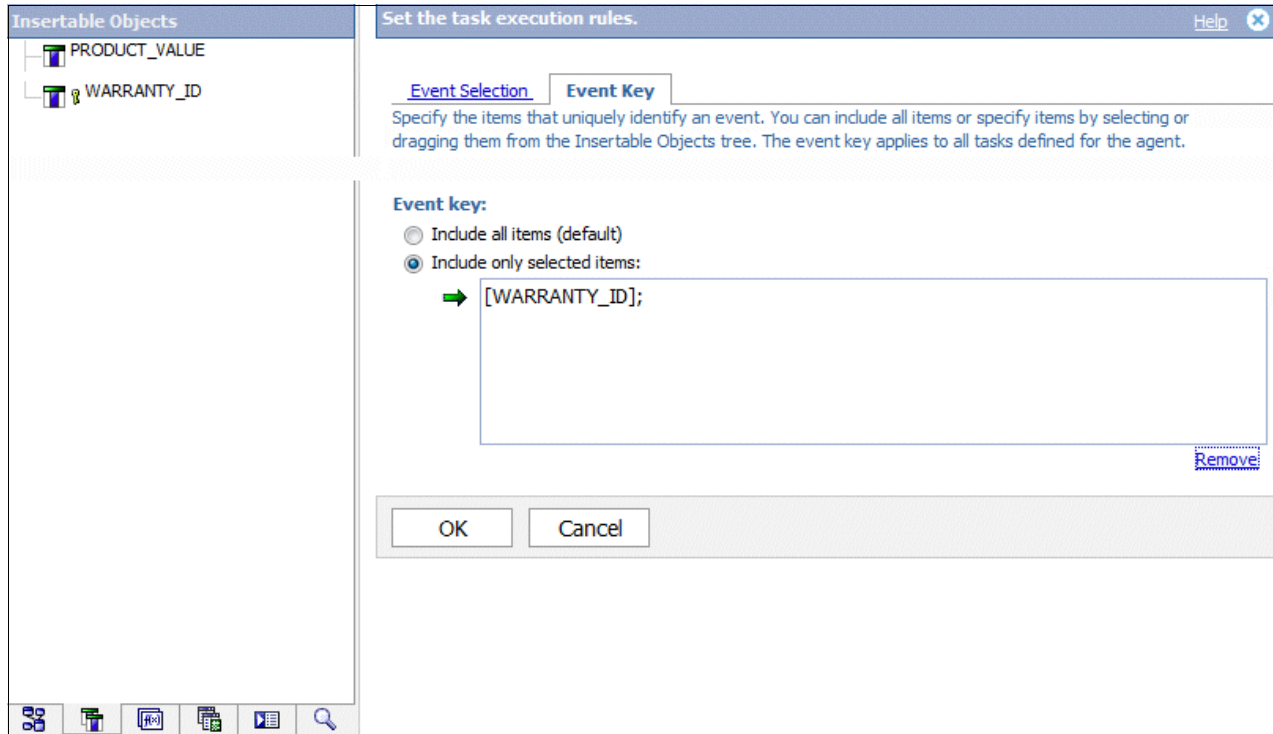


Figure 6-89 Event Key

5. Click **File** → **Save** to save your rule as Event for claims with product values ≥ 100 .
6. Select the location and then click **OK**.

6.6.4 Schedule the agent

Since you do not want to run your task manually, you can schedule it to run every hour, day, month, or year. Only one schedule can be associated to each agent. If you want to have multiple schedules for an agent, you create agent views and create a schedule for each view.

To create a schedule:

1. In the **I want to** pane, click **Schedule the agent**.
2. Select **Priority 1** from the Priority drop-down list.
3. Select the start time and choose **No end date** in the end time section (Figure 6-90 on page 234).
4. In the Frequency section, choose how often you want the agent to run to make the agent run daily; for example, click the **By Day** tab and set the agent to run daily every 2 hours, and then click **OK** when done.
5. Click **Save**.

Disable the schedule

Priority:

Start:

End: No end date
 End by:

Frequency:
 Select the frequency by clicking on a link.

Every minute(s)
 Every hour(s)
 Every day(s)

Daily Frequency:
 Every Hour(s) between and

Figure 6-90 Schedule the event



Testing with Rational Quality Manager

Testing is a key element of software development, and as such it requires careful planning to ensure a proper allocation of time and resources. Every time you create or modify software, you need to confirm that what is new works as designed. Since there is never enough time to cover all possible test scenarios, you also need to exercise due diligence to ensure your test process is planned, prepared, and executed properly.

You can no longer rely on ad hoc tooling, such as spreadsheets, to govern your testing activities; you need to have a collaborative environment that helps you communicate effectively with all testers across various teams while minimizing identified risks by supporting proper planning.

In this chapter we describe the required steps for testing the sample warranty claim system using the default test process defined in IBM Rational Quality Manager, which is a component of the Rational Solution for Collaborative Lifecycle Management (CLM) that was described in Chapter 2, “Customizing the methodology and creating a lifecycle project” on page 7. This tool is a centralized web test environment that helps you plan, develop, execute, monitor, and report your test efforts.

In addition, this tool summarizes how you can automate your testing using IBM Rational Functional Tester. IBM Rational Functional Tester, on the other hand, provides testers with automated testing capabilities for functional testing, regression testing, GUI testing, and data-driven testing. Using this tool, testers can automate their test scenarios for Java, web 2.0, SAP, Siebel, terminal-based and Microsoft Visual Studio .Net Windows Forms applications.

7.1 Test planning

When you start the test planning, the first thing you need to analyze is the nature of what you will be testing. Is it new? Is it an update? Does it integrate different technologies? Does it have to work on different platforms? What kind of users will access it? As you answer such questions, your aim is to confirm whether you are ready to release or not.

The integration of business process management and operational decision management in the warranty claims make it a complex system. The development of this application follows the process described in 2.2, “Customizing the methodology and the process” on page 8. This process produces functionality interactively. Therefore, at each phase where new functions are ready, they need to be tested to ensure these modules work together or defects are raised as needed.

Iterative testing requires reusable scenarios and flexible plans. Rational Quality Manager is an excellent option for an effective testing strategy.

7.1.1 Creating the test plan

A test plan is used to describe the test strategy for the application or solution being tested. A test strategy may include any combination of or all of the following aspects:

- ▶ Test objectives that define required results
- ▶ Entry criteria that summarize what is needed to start testing
- ▶ Test levels that confirm the depth of testing (that is, unit, functional, performance, accessibility, and security)
- ▶ Test environments that document the platforms, operating systems, browsers, and devices you test against
- ▶ Test schedules that articulate when and who will conduct the test phases
- ▶ Test cases that document the test procedures for each testing phase
- ▶ System requirements that define functions to be tested
- ▶ Risk assessments that identify testing prioritization
- ▶ Exit criteria that confirm when testing is complete

Once your test plan is approved, all team members will have a roadmap detailing how system quality will be verified. It is important to note that even though you have documented, reviewed and approved your test plan, it is subject to change over time. Therefore, your test plan should be easy to maintain and update.

If you use ad hoc tooling to create and maintain your test plan, you will experience keeping your document up to date. Generally, such tooling only allows single edits at a time, which creates a bottleneck when the updating becomes more frequent. You will want to have tooling that supports collaborative drafting, editing, updating and distributing of testing assets. Another important aspect to consider is that not all the projects require the same kind of test plan. A new release of a system requires more testing than a maintenance release. Therefore, you may need to have more than one template that can be easily adapted per project needs.

When you create a test plan in Rational Quality Manager, you can base your plan on a default template or you can create a new template. This flexibility makes creating test plans in Rational Quality Manager suitable for different teams that need to conduct one or numerous types of tests.

To create a test plan in Rational Quality Manager:

1. Select **Planning** → **Create Test Plan** from the dashboard.
2. Select the default template.

Note: We explain how to create templates in 7.1.2, “Creating templates” on page 240. For now, you can add or remove sections from the default template by clicking **Manage Sections**.

3. Name your test plan, select an owner and assign a priority. In the Summary section, you can define the plan categories you will use and update each to document your test plan.

Figure 7-1 shows a general view of a test plan.

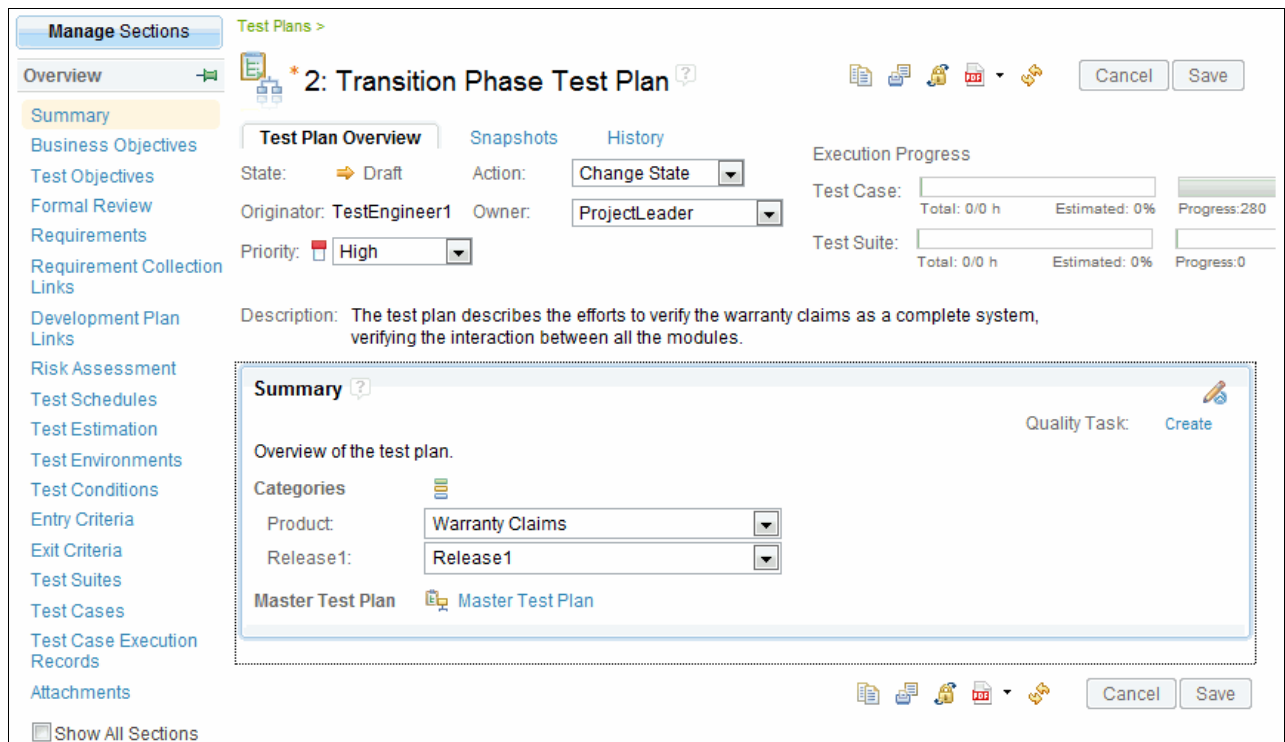


Figure 7-1 Creating the test plan

You can also link test plans, which allows you to create a master plan and individual plans for the project phases. In this example, we created a master plan and child plans for the construction and transition phases.

By default, the predefined templates included with Rational Quality Manager contain a section called Requirements and/or Requirements Collection Link. These sections are used to link the test plan to individual requirements or a collection. This linkage helps establish traceability.

To link a test plan with a requirements collection, click **Requirements Collection Link** and then click the green plus sign. Figure 7-2 on page 238 shows the dialog where you can search the collection.

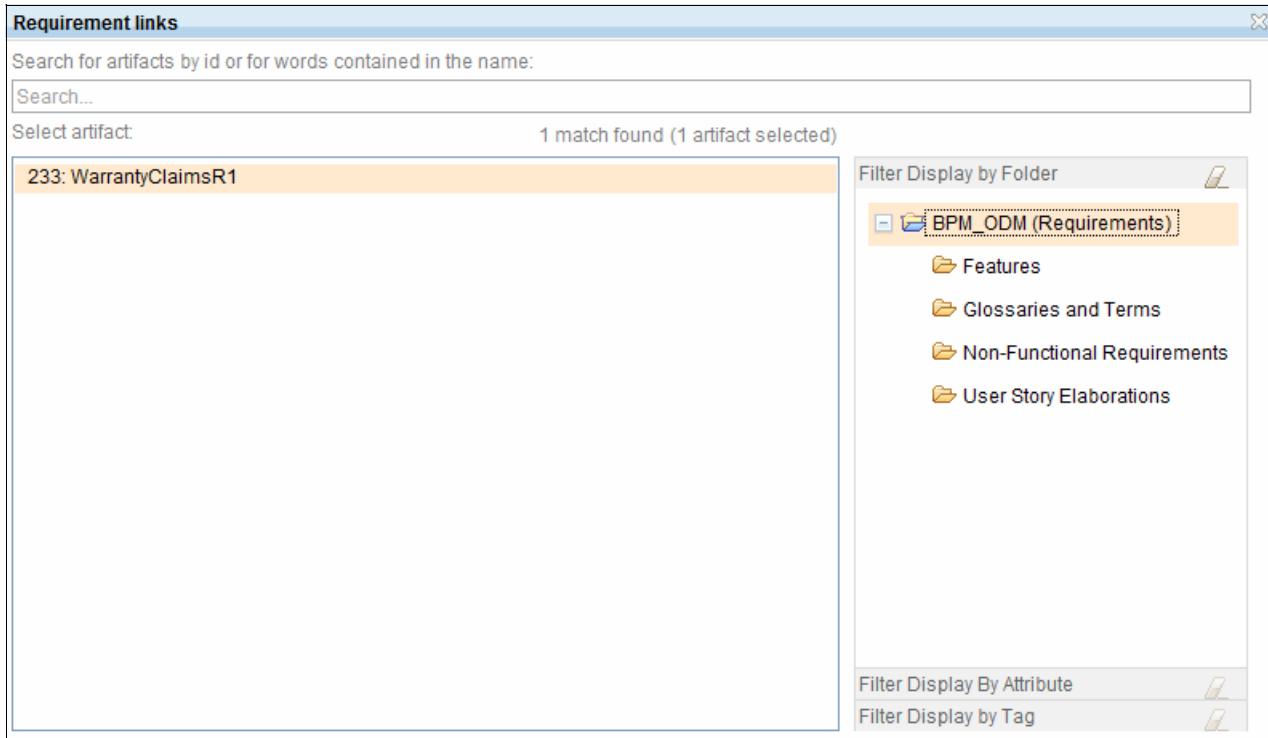


Figure 7-2 Linking requirements collection to a test plan

When the collection is linked, navigating to requirements is as easy as clicking the link. To access the requirements included in a collection without leaving Rational Quality Manager, just hover your mouse over the link to view the requirements in a pop-up dialog (see Figure 7-3). You can also open the collection in another browser tab or window.

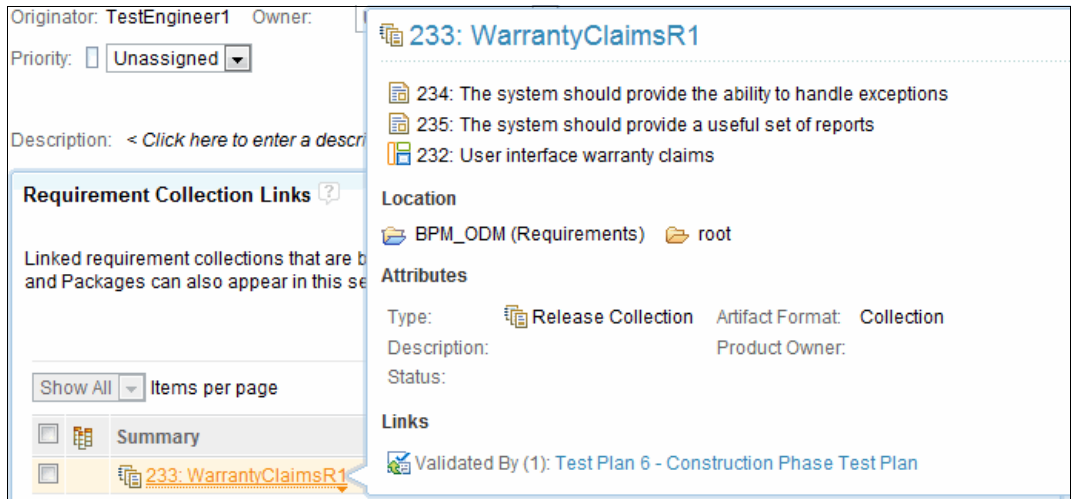


Figure 7-3 Hover the mouse over the collection link

You may also find it useful to link your test plan with your development plan. To do so, click the Development Plan Link and then click the green plus sign to select your Test Plan.

You need to define the supported test environments for this effort in which the testing is run to validate the system. To add the environments you should click the **Test Environments** link and you will get to define the Platform Coverage by clicking the pencil icon. After that, click the

Test Environment tab followed by the **Generate New Test Environments** icon to automatically create the required environments according to your platform coverage. See Figure 7-4.

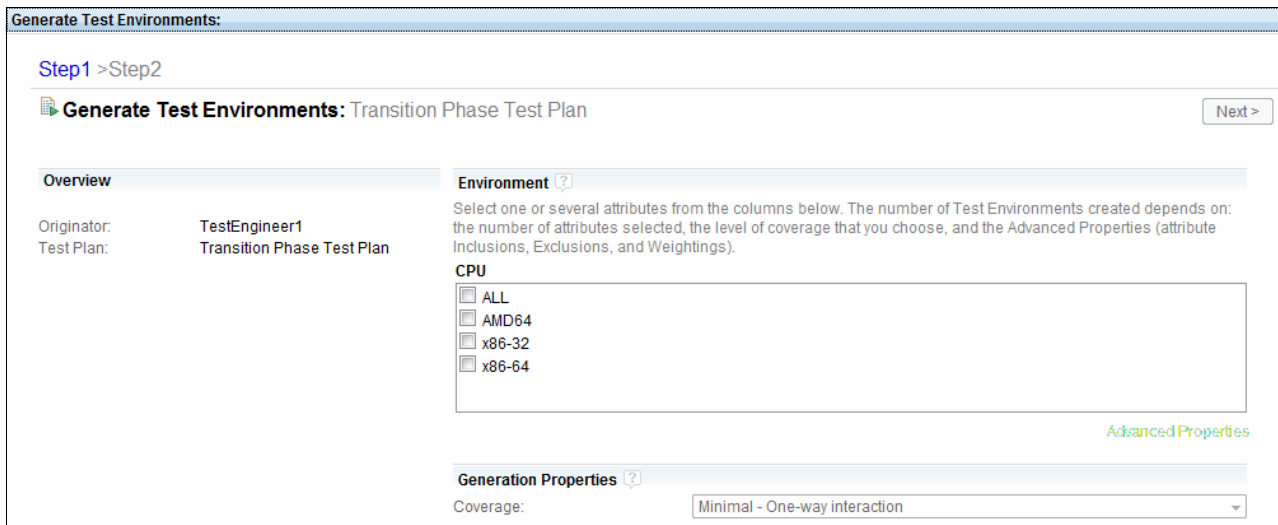


Figure 7-4 Generating test environments

If the test cases that are associated to this test effort exist when creating the plan, you can link them to your plan by going to the Test Cases link and then clicking the green plus link to add the applicable test cases. If they do not exist when the plan is created, they can be linked later by using these steps. They are also applicable for linking test suites.

A test schedule is critical for all test plans, and test engineers need to be prepared to confirm results by specified dates. A test schedule is derived from the project iterations.

To set up the test schedule in your plan:

1. Select the **Test Schedules** link and click the **Browse** link.
2. Select the iteration to include in your plan.
3. In the Action column click the **Plan Details** icon to assign the number of test points that should be completed per day of execution (S-curve); this detail will help the team confirm progress during testing.
4. Click the **Auto Generation** icon if you want Rational Quality Manager to split the points for you as shown in Figure 7-5 on page 240.

Test Schedules Quality Task: [Create](#)

Define the test schedule for this Test Plan. Browse and select an iteration to generate a schedule.

Show All Items per page Previous | 1 - 1 of 1 | Next

Test Schedule: Iteration: [Browse](#) [Clear](#)

Name	Description	Points	Planned Defects	Planned Start Date	Planned End Date	Planned Duration	Actions
Transition I1 [7/9/2012 - ...		0	0	Jul 9, 2012	Jul 21, 2012	13 d	

Execution Progress Planning

Plan Details

Date	Points attempted	Points completed	Comment
Jul 9, 2012	0	0	
Jul 10, 2012	24	9	
Jul 11, 2012	83	37	
Jul 12, 2012	202	103	
Jul 13, 2012	394	237	
Jul 14, 2012	632	453	
Jul 15, 2012	856	710	
Jul 16, 2012	1019	932	
Jul 17, 2012	1119	1077	
Jul 18, 2012	1174	1157	
Jul 19, 2012	1203	1197	
Jul 20, 2012	1218	1216	
Jul 21, 2012	1234	1234	

[Save](#)

Key Date:

Name *

No items found.

✓ Saved successfully

Figure 7-5 Test schedule

7.1.2 Creating templates

Templates are useful when configurations are constantly repeated. You can create templates for test plans, test cases, and test suites. Although in this section we do not cover the templates for test cases or test suites, the steps for creating these templates are similar to creating the test plan as described in 7.1.1, “Creating the test plan” on page 236.

The test plan template contains sections that you include or change when creating or editing a template. For example, you can create a template for a functional test plan, and then modify it to create another for your performance and regression test plans.

To create a template, perform the following:

1. Select **Planning** → **Manage Artifact Templates** from your dashboard.
2. A list of existing templates appears. Below the filter click the icon with a document and a plus sign.
3. A pop-up menu appears from which you select whether you want to create a test plan, test case, or test suite template; click the appropriate item.
4. Enter a name for the template and select the sections you want to include in the template from the Available Sections list.
5. Next, click the arrow to move your sections to the Selected Sections list and then click the **Set as new default** checkbox if you want all the new artifacts to use this format; click **OK** when done.

7.2 Test preparation

To create a useful set of test cases, you need to identify the required scenarios that can help you build strong suites. While working on this task, test engineers can identify those actions that will be repeated. These actions can be contained in a reusable test script, which will allow you to build flexible test cases.

7.2.1 Creating test scripts

A test script contains instructions to implement a test case. Test scripts can be manual, meaning a person executes the steps, or they can be automated with tools such as IBM Rational Functional Tester.

Manual test scripts

Manual test scripts can be created in Rational Quality Manager as follows:

1. Select **Construction** → **Test Scripts** in the Create section.
2. Specify the name of the test script.
3. Assign an owner.
4. Leave Manual as the type.
5. Enter a description for the script.
6. Click the **Manual Steps** section followed by the link named **Click to add step**.
7. Fill in the details for each step, or if you have any images, you can include them in each step by clicking **Insert image** on the editor.
8. Click **Save** to save your work.
9. Optionally, you can create tasks for other team members to provide details that will enable them to complete the test case.

Note: Whenever creating new items in your test plan, you will want to conduct the necessary reviews and obtain approvals per your governance process.

Figure 7-6 and Figure 7-7 on page 242 show a test script for the creation of a business rule in the IBM Decision Center portal.

The screenshot displays the Rational Quality Manager interface for creating a test script. The title bar shows the script name: "3: Creating new business rule based on text". The interface includes a "Test Script Overview" tab, "Snapshots", and "History" options. The "State" is set to "Draft", and the "Action" is "Change State". The "Work Item" is "110: Complete the Test Script : Creating new business rule based on text". The "Originator" and "Owner" are both "TestEngineer1", and the "Type" is "Manual". The "Test Data" is "Unassigned". The "Description" is "This script validates the BPM process developer can create new business rules in the IBM Decision Center". The "Summary" section includes "Categories", "Function" (Decision Center), and "Test Phase" (Construction Test). The "Formal Review" section is also visible, with a note to list reviewers and approvers.

Figure 7-6 Test script

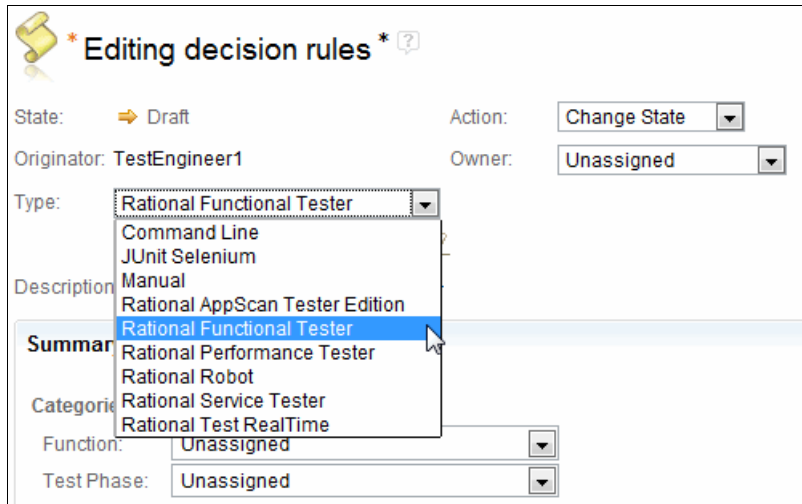


Figure 7-8 Selecting the type of script

To integrate Rational Functional Tester with Rational Quality Manager you need to configure the adapter that is installed by default with Rational Functional Tester. This adapter needs to be running on the computer where the test cases will be executed. Rational Quality Manager collects the results and reports test progress no matter where the testing was run from.

This adapter will connect automatically to Rational Quality Manager, so you will need to have a valid user ID. A Jazz administrator can create the needed ID and assign the user a required Rational Quality Manager - Connector license. For the sample the user is RFTadapter.

To start the adapter, navigate to the installation path where you installed Rational Functional Tester, or go to your operating system applications menu, and then select **IBM Software Delivery Platform** → **IBM Rational Functional Tester** → **Adapter to Rational Quality Manager** → **Configure Adapter**.

Provide the server address and the adapter user ID and select the project area, as shown in Figure 7-9.

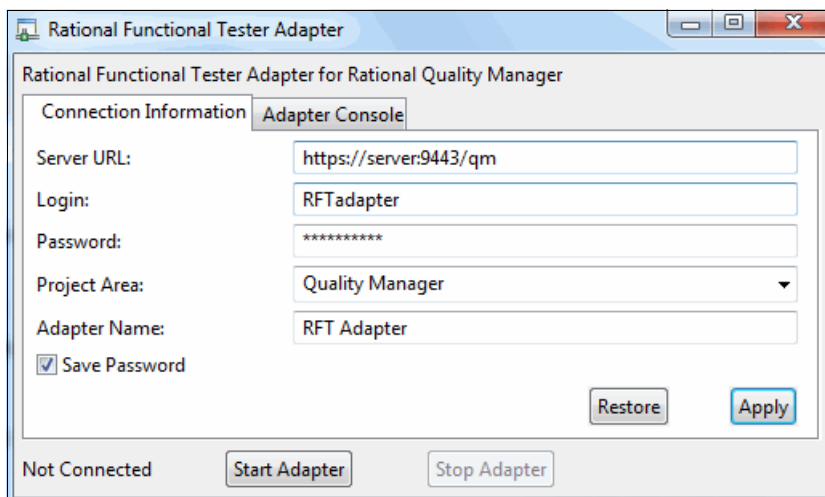


Figure 7-9 Rational Functional Tester adapter for Rational Quality Manager

Using Rational Functional Tester you can create automated and reusable test scripts for HTML, Java, Windows, .NET, Visual Basic, Silverlight, Eclipse, SAP, Siebel, Flex, Ajax, Dojo,

GEF, and PowerBuilder applications. Make sure you use Rational Functional Tester V8.2.2 or later if you are working with Rational Quality Manager v4.0.

Additional information: To get more information about Rational Functional Tester, go to:

<http://www-01.ibm.com/software/awdtools/tester/functional/index.html>

<http://www.redbooks.ibm.com/Redbooks.nsf/ibmpressisbn/9780137000661>

After automating your test scripts in Rational Functional Tester, you need to link them to Rational Quality Manager.

To create an automated script in Rational Quality Manager from an existing Rational Functional Tester script (Figure 7-10 on page 245):

1. Create a new test script by clicking **Construction** → **Test Scripts** in the Create section.
2. Provide the appropriate name, owner, and test phase.
3. Select Rational Functional Tester as type.
4. Click the pencil icon to edit, select the appropriate radio button to confirm where your test scripts reside: either locally or on a remote system, and then click **Select Adapter**.

Note: The test scripts created for the sample application reside on a local machine, so we selected **Use test resources that are local to a test machine**.

5. Choose the adapter belonging to your local computer and click **Next**.
6. Enter the path where your scripts are located and click **Finish**.
7. If required, you can provide additional details via arguments to confirm how the scripts should be run,
8. Press Ctrl-s to save your test script.

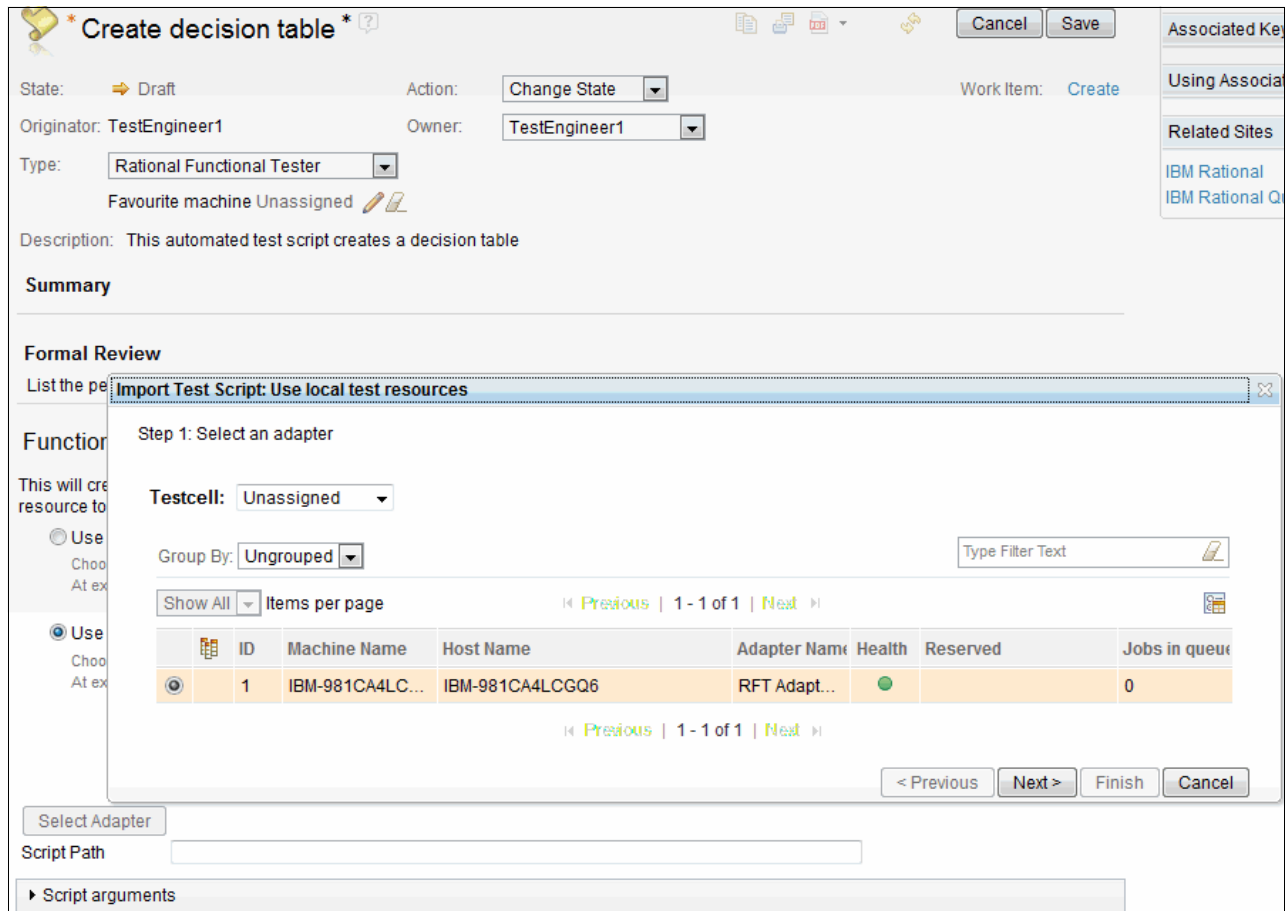


Figure 7-10 Automated script created with Rational Functional Tester

Your automated test scripts can run in a test cell, which is a group of lab resources that describe an environment. For example, a test cell can contain a computer running the Rational Functional Tester adaptor.

Manual test scripts generated by a manual test script recorder

You can use the manual test recorder feature of IBM Rational Quality Manager to simplify the creation of manual test scripts.

Note: The application that you are testing must be an HTML application, a Java application, or an application with Dojo controls.

The manual test recorder utilizes an integration with IBM Rational Functional Tester that lets you automate the creation and editing of manual test scripts. You use a manual test script recorder to record user gestures in your application, which are then converted to manual script statements in the English language. In addition, the manual test script recorder is capable of capturing screenshots from the test application where user gestures were made.

Using the manual test script recorder to capture script steps saves your team time and eliminates user errors that might occur when writing manual test script steps. You can also use the manual test script recorder for exploratory testing.

The basic workflow for configuring and using the manual test recorder follows:

1. Configure the Rational Functional Tester adapter for use with IBM Rational Quality Manager.

Note: By default, the test environment is automatically enabled by Rational Functional Tester. In certain scenarios, however, you may need to enable specific components, such as browsers, Java runtime environments (JREs), and Eclipse platforms, to support your testing objectives.

2. Start the Rational Functional Tester adapter, and as needed, modify user preferences for the manual test script recorder.
3. Create or modify a manual test script with the manual test script recorder. Figure 7-11 shows the Record Manual Script dialog that opens when the recorder is started.

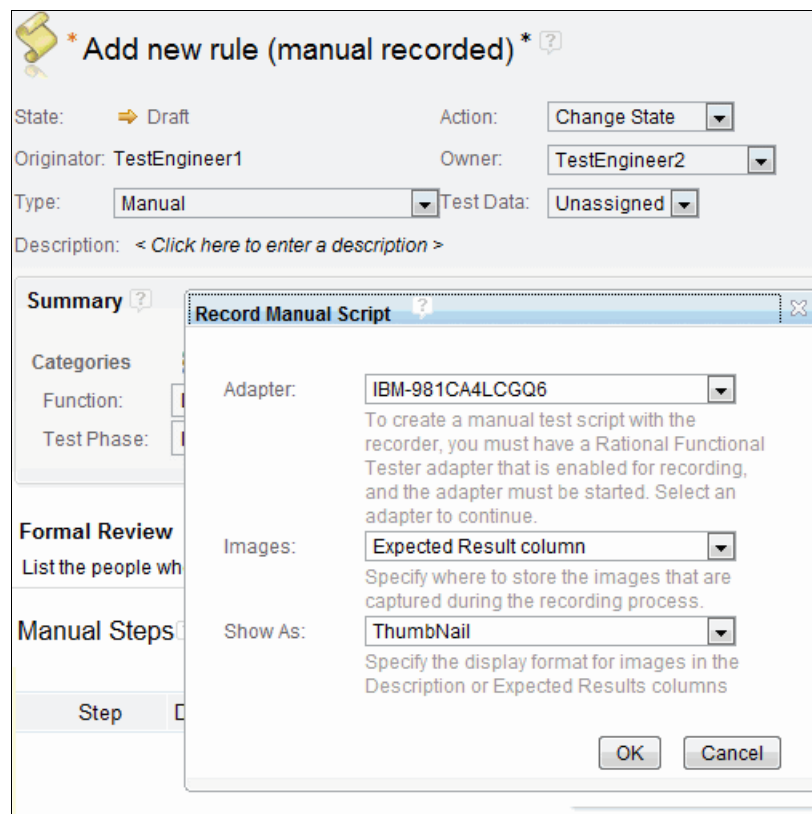


Figure 7-11 Recording manual script

Additional information: For more information about the manual test script recorder, go to:

http://pic.dhe.ibm.com/infocenter/clmhelp/v4r0/topic/com.ibm.rational.test.qm.doc/topics/t_construct_scripts.html

7.2.2 Creating test cases

You develop test cases to define the things that you must validate to ensure that the system is working correctly and is built with a high level of quality. Each test case is typically associated with a test script, although you can run a test with no associated test script.

You can also associate more than one test script with a single test case. Each test script associated with a test case should provide a different way to test the scenario in the test case. For example, a test case might require several scripts to test the scenario in different test environments. Or, you might have both manual and automated scripts that can be used to exercise the test case.

If you find that a test case actually represents a larger scenario, break it out into separate test cases, group them in a test suite, and associate test scripts with each test case in the suite. Do not use test cases to group test scripts that exercise different aspects of a common function. Instead, use test suites to group related test cases that you want to execute together. With a test suite, you can plan, initiate, and track the execution of the related test cases that make up the larger scenario that is tested by the test suite.

To create a test case follow these steps:

1. From the main menu, select **Construction**, and then click **Test Case** from within the Create section.
2. At the top of the new test case window (a table of contents is on the left and an editor on the right), enter a name for your new test case.

Note: If team areas are enabled for your project area, select a team area from the list; you must explicitly enable team areas in Manage Project Properties.

3. Select a test case template from the list, and then select an owner and a priority for the test case.
4. Next, add a description.
5. You can complete the test case Summary section or assign the Summary to another team member to complete as follows:
 - You can use categories to define a hierarchical organization of test plans, test cases, test suites, test scripts, and other test artifacts. You can also define subcategories and other category relationships. You can use the default categories or create your own by clicking the **Manage Test Case Categories** icon.
 - Type a number to estimate the amount of time required to complete your test case and select a time interval for the estimate. For example, if you type 1.5 in the Estimate field and select days for the interval, the estimate is represented as "1 day 4 hours."
 - You can use weight as a means of tracking the relative difficulty or importance of your test case. This weight is provided as a measurement of results, not a measurement of time. When a test case is executed, the weight is calculated based on the success of the steps included in the test script.
 - If an administrator defined any custom attributes for test cases, they display in the Attributes section of the Summary. The field values for custom attributes can be in the format of text, integer, or date, and you can sort or filter your test cases using custom attributes in the same way that you use categories.
6. Click **Manage Sections** to add your own, customized test case sections or to remove unneeded sections.

If you use the Collaborative ALM Test Case template, you will have the Development Items and Requirements Links included. You can link your test case to the requirement and to the development task in order to keep full traceability of the assets. See Figure 7-12 on page 248, Figure 7-13 on page 248 and Figure 7-14 on page 248.

7. Click **Save** when done.

Note: At any point after you create the test case, you can add it to one or more test plans.

1: Creating decision rules

State: Draft Action: Change State

Originator: TestEngineer1 Owner: TestEngineer1

Priority: Medium

Description: This test case verifies the creation of decision rules

Summary

Use the theme, category and function features to group your test cases along related items or logical groupings. Weight is a measure of execution effort and can be based on tester hours or units of work.

Quality Task: Create

Categories

Function: Decision Management

Test Phase: Construction Test

Estimate:

Weight* 40 Points

Figure 7-12 Test case

Development Items

Quality Task: Create

Change management items that are aligned with the testing

Type Filter Text

Show All Items per page Previous | 1 - 1 of 1 | Next

Summary

108: Implement Rules

Previous | 1 - 1 of 1 | Next

Figure 7-13 Linking development tasks to test cases

Requirement Links

Quality Task: Create

Linked requirements that are being validated

Type Filter Text

Show All Items per page Previous | 1 - 1 of 1 | Next

Summary

233: The system should provide an automatic mechanism to make decisions about claims based on ...

Previous | 1 - 1 of 1 | Next

Figure 7-14 Linking requirements to test cases

8. Include the test scripts as shown in Figure 7-15.

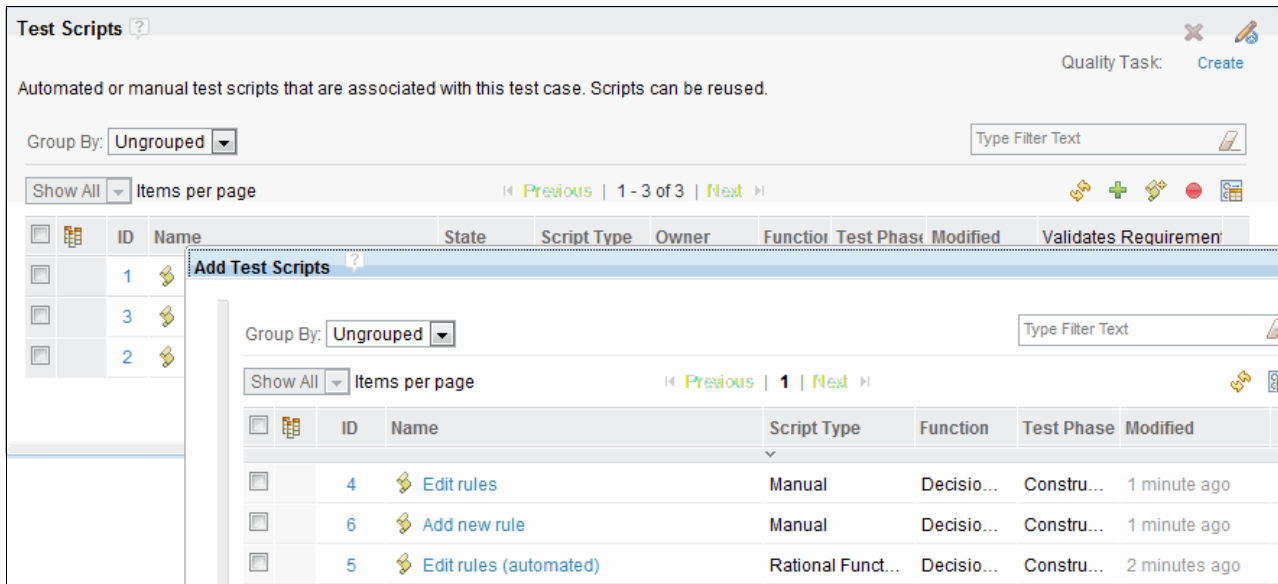


Figure 7-15 Linking test scripts with test cases

7.2.3 Creating test suites

Test suites are used to group a collection of test cases for test execution purposes. You can associate new or existing test cases with the test suite.

Creating a test suite is very similar to creating a test case except that when adding test cases you can specify whether you want to run the test cases in sequence or in parallel, which is useful for test cases that contain automated scripts.

1. In the main menu, click **Construction** → **Create** → **Test Suite**.
2. At the top of the new test suite window, enter a name for the new test suite.
3. If required, select a team area from the list.
4. Select a test suite template from the list, followed by an owner and a priority for your test suite.
5. Type a description and complete the test suite Summary section.
6. Click **Manage Sections** to add your own, customized test case sections or to remove unneeded sections.
7. Click **Save** when done.

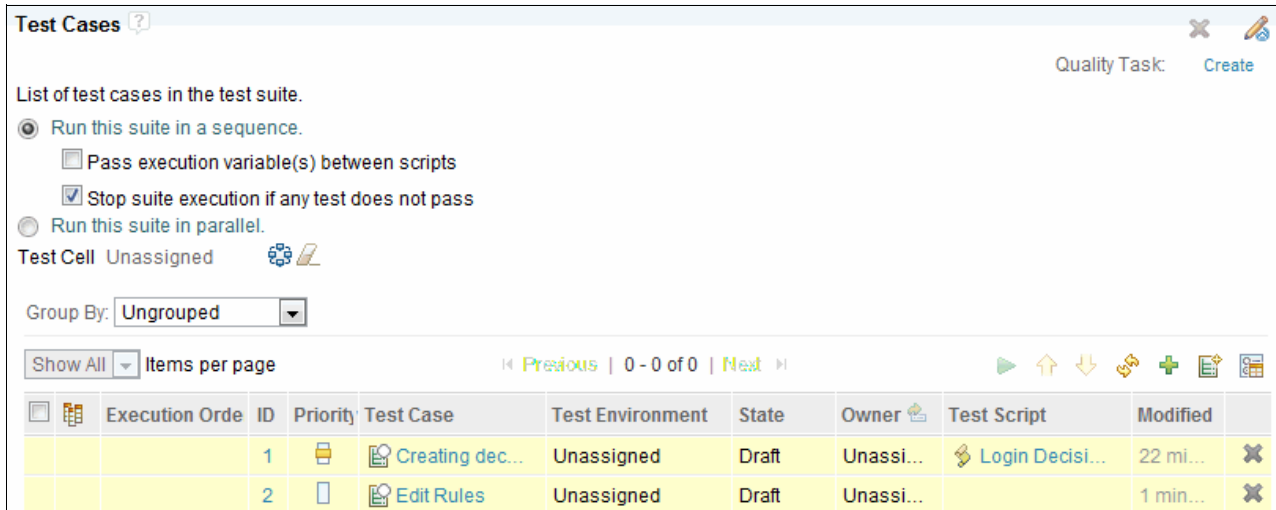


Figure 7-16 Test cases included in a test suite

7.2.4 Storing reusable test data

In IBM WebSphere Operational Decision Management you can use the Decision Validation Services to test and simulate rulesets in Rule Designer and Decision Center.

To correctly implement and use Decision Validation Services, do the following:

- ▶ Enable the ability to run tests and simulations from Decision Center and remote testing from Rule Designer.
- ▶ Customize testing and simulation features for your specific needs.

Testing and simulation in Decision Validation Services give users the ability to validate usage scenarios to ensure correctness of rulesets, or to see the potential impact of changes to rules.

Central to both testing and simulation, scenarios represent real or fictitious input data to rule execution, and are stored in Microsoft Excel or another format when appropriate. The scenario file template contains columns with no data for all required input data. To validate your rules, populate the file with data for all necessary scenarios. Your completed scenario file is used to run the test.

During testing or simulation, you submit scenarios to a Scenario Service Provider (SSP), either locally in Rule Designer or configured in Rule Execution Server for remote testing. In either case, the SSP returns a report of the results of rule execution on the scenarios. You can store your test data in Rational Quality Manager to generate your testing inventory, or you can convert your Microsoft Excel file to a comma-separated values (.CSV) file via the save function for you to import it to Rational Quality Manager for reuse. See Figure 7-17 on page 251.

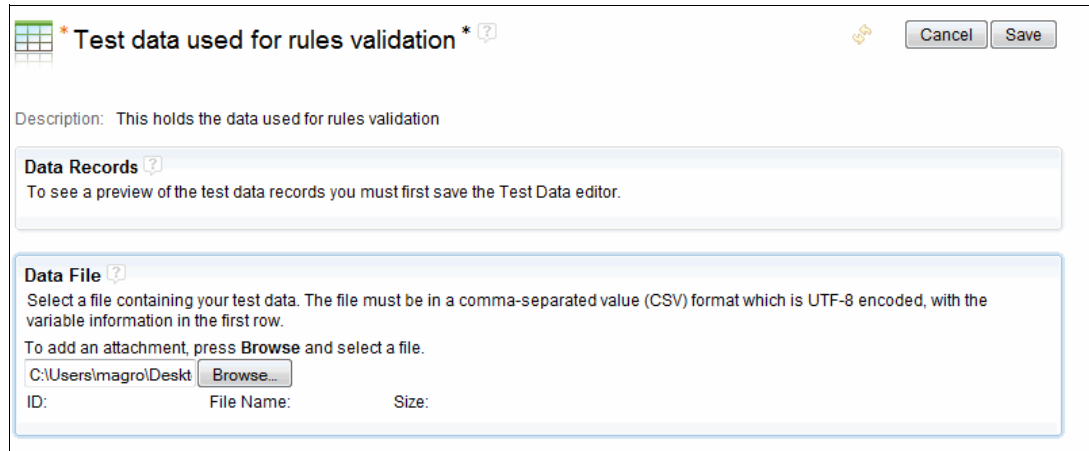


Figure 7-17 Test data

For more information about Decision Validation Services, go to:

http://pic.dhe.ibm.com/infocenter/dmanager/v8r0/topic/com.ibm.wodm.dserver.rule.s.overview/topics/tpc_testing_dvs.html

7.2.5 Creating the test execution records

Test case and test suite execution records map the planning and test environment information to the respective test case or test suite. These records specify the hardware and software to complete the required testing. They basically define a test this way: run the test case or test suite on this hardware platform, running this operating system, using this browser (that is, software).

To ensure that your test case or test suite passes when it runs in four different test environments, you can create a test case execution record or test suite execution record for each test environment. Test case execution records and test suite execution records also contain the overall result (pass, fail, blocked) that is associated with the execution of a test case or test suite.

You can create test case and test suite execution records individually or use a wizard to generate several at once automatically. Before you can generate test case or test suite execution records, you must have created at least one test case or one test suite.

The high-level process for generating a test execution record follows:

1. Go to the test plan that should have the test cases associated already.
2. Go to the **Test Cases** link.
3. Click the checkbox to select the test cases that require test execution records.
4. Click the **Generate Execution Record** icon as shown in Figure 7-18 on page 252.

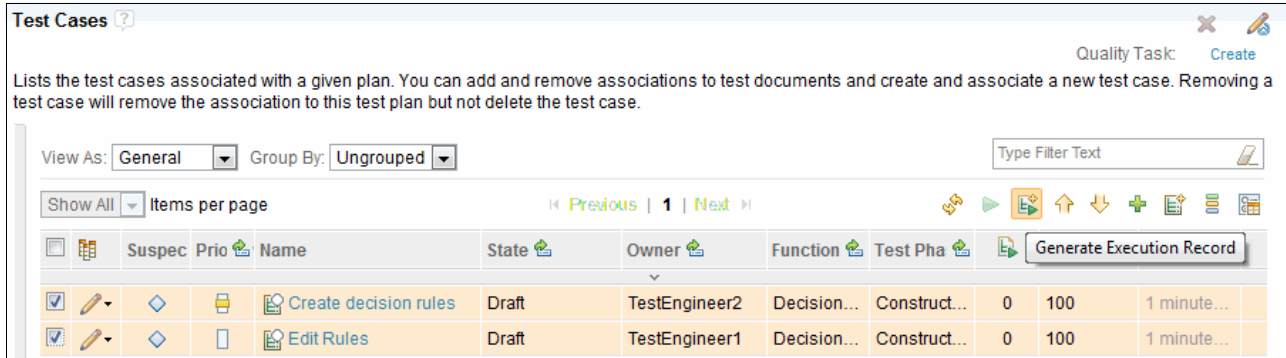


Figure 7-18 Adding execution records from the test plan

7.2.6 Test lab preparation

Once they receive their assigned test execution records, your test engineers, or a lab administrator if this role exists in your organization, will know what test environments are needed to complete the required testing. They can proceed with setting up the physical and virtual environments to accommodate the defined testing schedule.

Rational Quality Manager can be used by lab administrators to provide lab management capabilities. Your administrator can create the environments in the tool for your test engineers to request when needed to complete their testing. These capabilities can help reduce downtime and delays, because testing resources can be allocated more efficiently to meet the demands of the testing schedule.

Figure 7-19 on page 253 shows how easy assigning testing environment can be through Rational Quality Manager.

Request <07:05:36> * ?

Created by: TestEngineer1 Status: Draft Priority: High

Assigned to: Unassigned Respond by: Jun 20, 2012

Description: * Need test environment for integration test

Reservation

Enter the date range for the requested resources.

From: Jun 21, 2012 12 00 AM

To: Jun 29, 2012 11 59 PM

Lab Resources

Specify the test environments for the request.

Enter a label for the Lab Resource Description 1 ✖

Type: +

Operating System: + ✖

Figure 7-19 Lab request

7.3 Test execution

During test execution, test engineers will go through their assigned test execution records to perform their assigned actions. If the test cases or scripts are automated, the test engineer would execute each by specifying the environment (that is, hardware and software) to run on. To start the execution, test engineers follow these steps:

1. Go to **Execution** → **Test Execution Records**.
2. Use the filters to find specific records.
3. Select the record or records to execute and click the play icon. See Figure 7-20 on page 254.

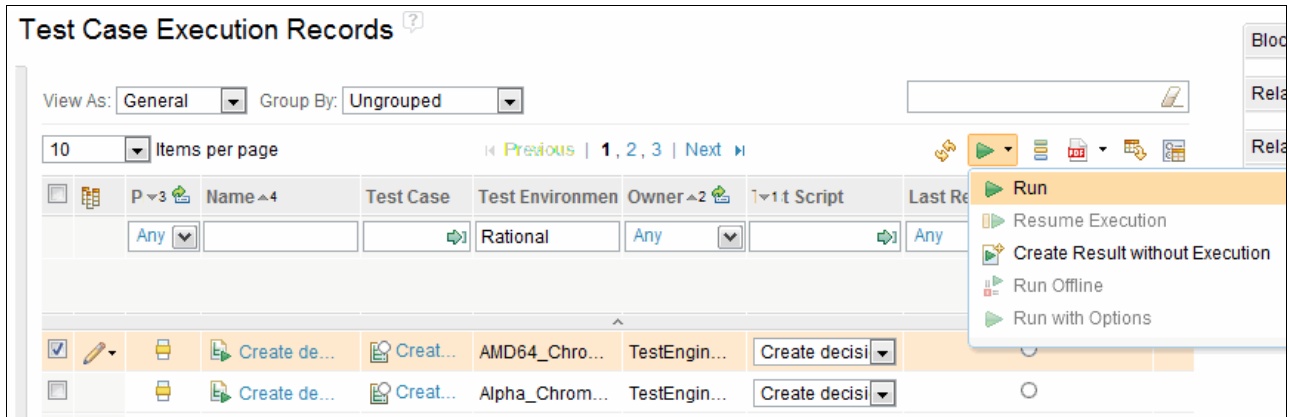


Figure 7-20 Starting execution

During test execution, you can save panels to show the test results for each step if needed for audit purposes. When all the test steps have been completed, a dialog displays the test results. See Figure 7-21.

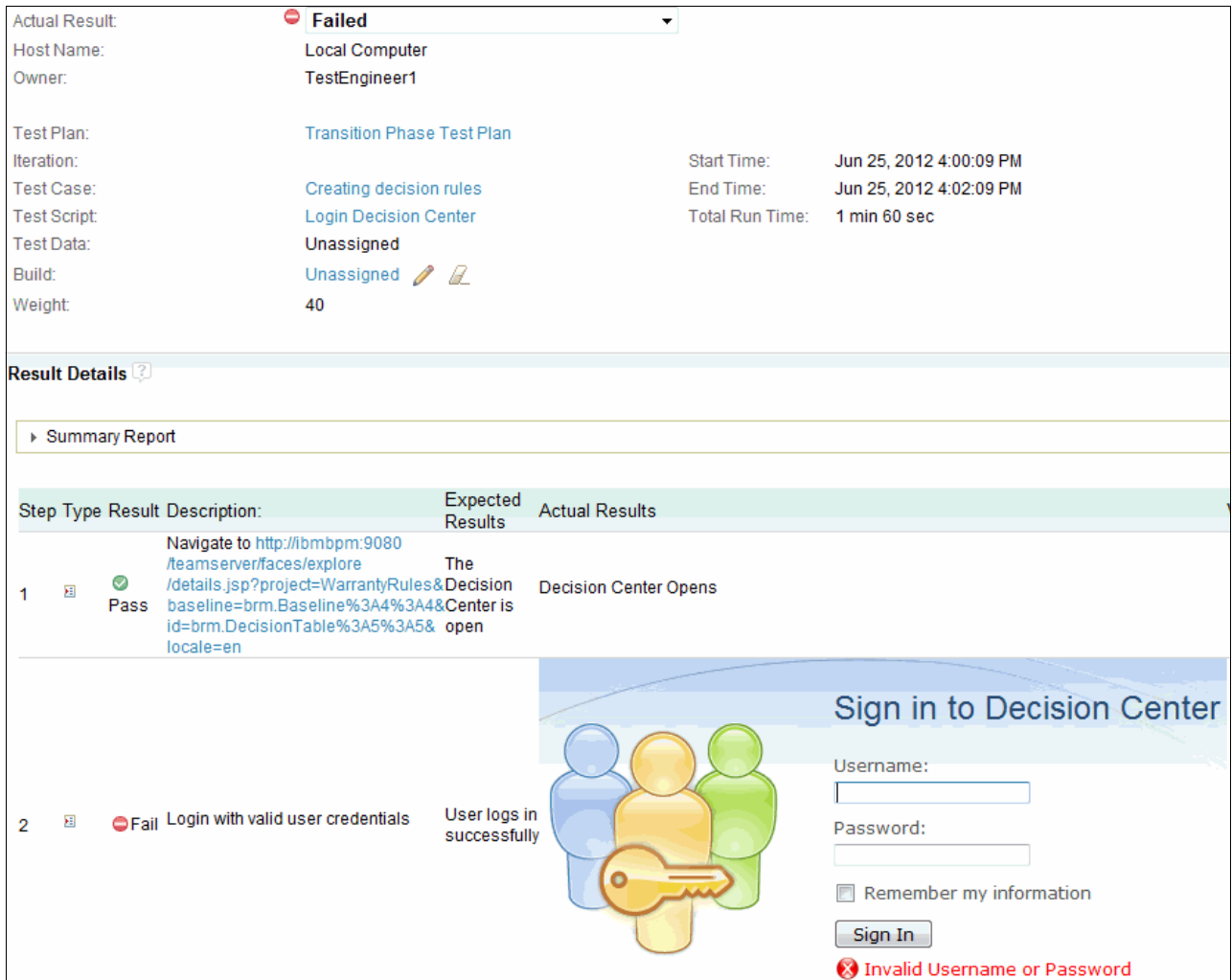


Figure 7-21 Test results

On the top of the test results, there is a button named Show Result Page. Click it to view all the test execution details.

If you click the **Defects** link in the left menu, you can create a defect record and associate it to the current test result. You may also link it to an existing defect. If you are using Rational Team Concert and have it integrated with Rational Quality Manager, your developers will see this new defect in their defect queries in Rational Team Concert. See Figure 7-22.

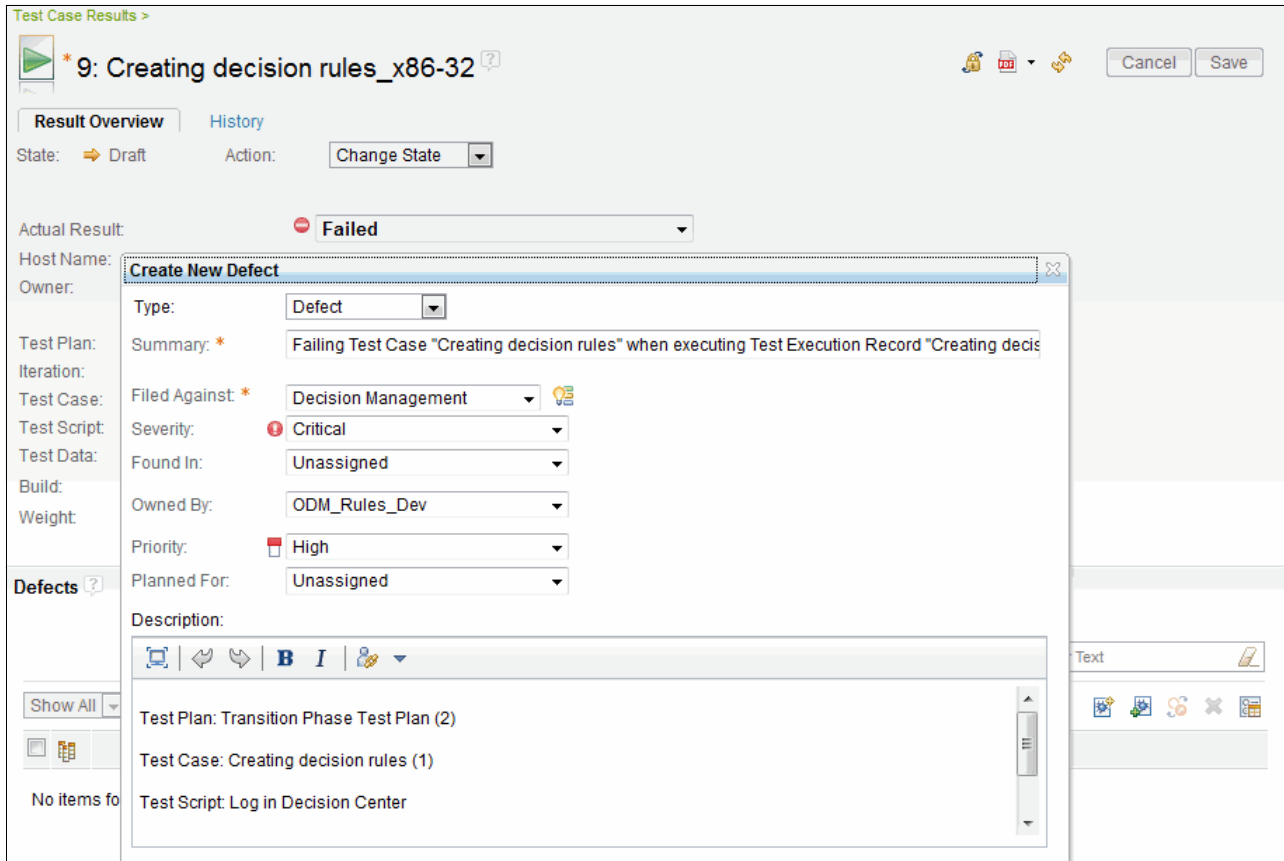


Figure 7-22 Creating defect from Rational Quality Manager in Rational Team Concert

The integration of Rational Quality Manager and Rational Team Concert helps streamline the defect management process, because the developers can now begin working defects assigned to them via the tooling, instead of awaiting alternate notifications from the test engineers.

Figure 7-23 shows how the defect is reflected in Rational Team Concert.



Figure 7-23 Defect in Rational Team Concert

Once the defect is fixed, the test engineer can execute the test case again to validate that the fix works. All these testing results are stored to provide auditability. See Figure 7-24

Previous Results

Group By: **Actual Result**

Show All « Previous | 1 - 4 of 4 | Next »

	Start Date	Start Time	Total Run Time	Script Name	Script Type	Owner	Execution Result
Failed (1 - 2 of 2)							
	Jun 25, 2012	4:00 PM	1 min 60 sec	Login Decisio...	Manual	TestE...	
	Jun 19, 2012	9:39 AM	5 min 11 sec	Login Decisio...	Manual	TestE...	
Passed (1 - 2 of 2)							
	Jun 25, 2012	6:20 PM	11 sec	Login Decisio...	Manual	TestE...	
	Jun 19, 2012	10:11 AM	12 sec	Login Decisio...	Manual	TestE...	

Figure 7-24 Previous execution results

When you use automated scripts, you can also create an automated schedule to run a suite. This makes sense for smoke or regression testing that must be completed several times. Rational Quality Manager can start the execution automatically and the test engineer can review the results at a predefined time. See Figure 7-25.

Execution Schedules >

Smoke test execution schedule

Originator: TestEngineer1 Owner: ProjectLeader

Description: This is the schedule that will automatically run the daily smoke test

Overview

Schedule an execution based on:

Date and time:

Start Date: Jun 20, 2012 12:00 AM

Repeat: Daily Repeats Every: Any Repeat Until: Jul 21, 2012

Next Scheduled Time:

Build completion:

Select Build Definition: Unassigned

Start execution schedule based on build status: Success (OK)

Not Scheduled Yet

Test Cell: Smoke test cell

Steps

Steps of the execution schedule:

Show All « Previous | 1 - 1 of 1 | Next »

No.	Step	Executable	Target	Status
1	Test Suite Execution Record	Smoke Test_Linux_Firefox	Multiple	

Figure 7-25 Execution schedule

7.4 Test traceability

Traceability is used to trace a project element to related project elements, especially those related to requirements. Traceability helps determine that a requirement is satisfied from inception through implementation and testing. You create traceability relationships by linking requirements to other requirements and to development and test artifacts.

Traceability supports these activities:

- ▶ Derivation, elaboration, and coverage analysis - Trace between requirements to specify the source, derivation, or dependencies between the requirements; you use traceability to verify that all requirements have the required elaboration to fully describe the objectives.
- ▶ Impact analysis - Assess the impact of change, or proposed change, by using traceability; it can be the basis for negotiating the change, because it helps expose dependencies and "hidden" costs of the change.
- ▶ Lifecycle coverage analysis - Verify that all the necessary development and test items are in place for implementing and validating requirements.
- ▶ Progress analysis - Track progress and completion status of linked requirements, development items, and test cases.

Rational Quality Manager also provides different views for traceability (Figure 7-26). Tracking test progress and traceability is easy by configuring the quality dashboard. In Rational Quality Manager V4, an execution tab is created by default so you just need to configure the reports to have a quick and easy view of the test progress.

ID	Name	Validates Requirement Collection	Tests Development Plan
2	Transition Phase Test Plan	219: WarrantyClaimsR1	Release Plan
1	Construction Phase Test Plan	219: WarrantyClaimsR1	Construction Plan - It...
3	Master Test Plan	--	--

Figure 7-26 Test plans traceability view

Test cases are linked to requirements and development tasks. See Figure 7-27.

ID	Name	Validates Requirement	Tests Development Item
1	Creating decision rules	233: The system sh...	108: Implement Rules

Figure 7-27 Test cases traceability view

Test execution records are linked to defects. See Figure 7-28 on page 258.

Name	Test Case	Defects
Edit Rules_x86-64	Edit Rules	112: Failing Test Case "Edit Rules" w...
Creating decision rules_x...	Creating decision rules	--

Figure 7-28 Test execution records traceability view

7.5 Other useful IBM tools related to the topics in this chapter

IBM Rational AppScan® Tester Edition helps to find and remediate security vulnerabilities, and it can be integrated with Rational Quality Manager. This tool can be a useful addition when developing the sample warranty application, as you can implement security testing in the development phase to identify potential issues early in the cycle, which can save time and money. For a complete and effective solution, your security testing tools should be integrated with your development workflow.

To learn more about Rational AppScan Tester, go to:

<http://www-01.ibm.com/software/awdtools/appscan/tester/>

For details about integrating it with Rational Quality Manager, go here:

http://publib.boulder.ibm.com/infocenter/rqmhelp/v1r0m0/topic/com.ibm.rational.test.qm.doc/topics/c_appscan_adapter.html

IBM Rational Performance Tester is used to identify the presence and cause of system performance bottlenecks. It integrates with Rational Quality Manager through an adapter, which serves as a bridge between the test tool and Rational Quality Manager. The adapter listens to requests from Rational Quality Manager and then runs tests on the test machine where the test tool is installed.

Rational Performance Tester generates performance and throughput reports in real time, enabling you to detect performance problems at any time during a test run. These reports provide multiple filtering and configuration options that can be set before, during, and after a test run. Users can also review performance trends from one build to another. Additional reports are available at the conclusion of the test run to perform deeper analysis on items such as response time percentile distributions.

As with Rational AppScan Tester, Rational Performance Tester can help you save time and money by integrating your performance testing tooling with your development workflow.

To learn more about Rational Performance Tester, go here:

<http://www-01.ibm.com/software/awdtools/tester/performance/>

For details about integrating it with Rational Quality Manager, go here:

http://publib.boulder.ibm.com/infocenter/rqmhelp/v1r0m0/topic/com.ibm.rational.test.qm.doc/topics/c_rpt_adapter.html



Deployment options with Workload Deployer

One of the biggest inhibitors in both development and test is the ability (or lack thereof) to acquire, deploy, configure and host environments to perform development, prototyping, and unit and full product testing. Cloud computing enables development and test teams to create and host such environments rapidly, especially when cloud resources can be managed and deployed using IBM Workload Deployer.

In this chapter we provide a brief overview of how IBM Workload Deployer can be used to deploy virtual system patterns to a private cloud environment. It describes the concepts, tooling and requirements for this type of deployment, as follows:

- ▶ IBM Workload Deployer overview
- ▶ IBM Workload Deployer patterns
- ▶ Administrative interfaces
- ▶ Appliance settings
- ▶ Tools for building custom assets
- ▶ Deploying virtual patterns to the cloud
- ▶ Applying a virtual application pattern to our scenario
- ▶ Additional information

We conclude the chapter by showing how IBM Workload Deployer users can create the requisite virtual images for each of the products used in this book to complete a virtual application pattern for deployment.

8.1 IBM Workload Deployer overview

IBM Workload Deployer provides a solution for creating, deploying and managing workloads and resources in a private cloud. It allows you to build and deploy virtual systems quickly, consistently and repeatedly, all from configurable virtual images. It also provides an application-centric capability that supports rapid deployment of complex business applications to development, test, or production environments.

8.1.1 Features and benefits

IBM Workload Deployer is based on the IBM DataPower® 7199/9005 product family. This appliance provides several benefits:

- ▶ **Consumability**

IBM Workload Deployer is available for use after accepting the user license agreement and performing minimal setup.

- ▶ **Security**

IBM Workload Deployer manages a shared, multitenant environment, where isolation and security are of utmost importance. The secure nature of this appliance is rooted in a self-disabling switch, which gets triggered if the appliance cover is removed. This physical security allows IBM Workload Deployer to serve as a secure vault for credentials, which can be tied to virtual images throughout their entire lifecycle (for example in storage, being deployed, running in the cloud, or being removed from the cloud).

- ▶ **Storage**

IBM Workload Deployer contains a storage driver that streamlines the maintenance of virtual image customizations. When an image is loaded onto the appliance, it is “shredded” into parts by the storage driver. When a virtual image is later customized and reloaded to the appliance, it is similarly shredded in a consistent and deterministic way. These collections of shredded images are then compared and only the newer or modified image is stored, thereby reducing the total storage requirements of your image repository.

- ▶ **Performance**

IBM Workload Deployer serves as a dedicated store for both preloaded and customized middleware virtual images and patterns. The appliance includes advanced compression and storage techniques that enable a user to store a significant number of these sizeable virtual images. It is also backed up by the DataPower processing power, which streamlines the provisioning and management of these images.

- ▶ **Cost**

The total cost of ownership (TCO) that is associated with a physical appliance is low. Over its life, an appliance requires less updating by specialized skills. Further, functionality and performance are managed at the system level, thereby reducing the day-to-day skill requirements.

8.1.2 Cloud types

IBM Workload Deployer is one of the foundational elements of the IBM private cloud strategy, mainly due to the degree of control it offers users. Private clouds are cloud services that are provided within an organization under its full control, while public clouds are cloud services provided and maintained by third parties outside of an organization's control. Spanning both are Hybrid clouds, which leverage services that reside in both private and public spaces. See Figure 8-1 on page 261.

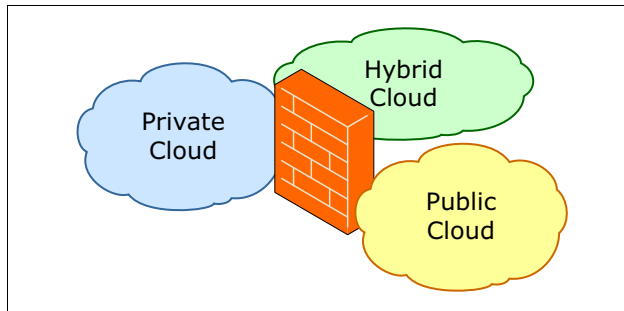


Figure 8-1 Cloud types

IBM Workload Deployer provides a proven path for leveraging existing IT resources by improving the efficiency and flexibility of these physical assets. It can enable organizations to become more responsive to changing business and market conditions, because development, test and production systems can be deployed and running in substantially less time when compared to traditional deployment methods.

8.1.3 Solution elements

The IBM Workload Deployer provisions both standard and customized virtual images, which may consist of middleware, applications, or complete solutions. With regard to middleware images, these can be preloaded on the appliance to provide building blocks for organizations to develop, test and deploy business applications easily and quickly.

Note: IBM Hypervisor Edition software products are optimized for virtualized environments, enabling higher hardware utilization and significantly faster time to value for deploying and configuring environments. These products are built using known, stable, and tested configurations.

IBM Workload Deployer can receive and act upon operational data from private cloud resource pools (IBM PowerVM®, IBM z/VM®, VMware hypervisors) to recover from hardware failures, image faults, and so forth. It can also monitor application workload demand conditions and adjust resource allocations in order to achieve established service level agreements for virtual applications.

When the virtual environment is no longer needed, its resources can be returned to the shared resource pool automatically and logged for internal charge-back purposes. IT managers can set access controls to system resources for individual users or groups, which optimizes the efficiency of virtual environments.

IBM Workload Deployer integrates seamlessly with development and service management tools from IBM Rational and IBM Tivoli® to provide enterprise-level architectural, design, development, management, and monitoring capabilities.

Figure 8-2 on page 262 shows a virtual platform that consists of a physical appliance with its hardware configuration and management application firmware, administration interfaces, preloaded and configurable middleware virtual images, configurable patterns and script packages. There is the private cloud environment on which the virtual systems and applications will run. This cloud environment consists of hypervisors, networking infrastructure, and storage devices that are allocated to the appliance. The virtual system images are deployed by the IBM Workload Deployer appliance into the cloud using the

intelligent placement capabilities, which guarantee efficient cloud resource usage coupled with high availability.

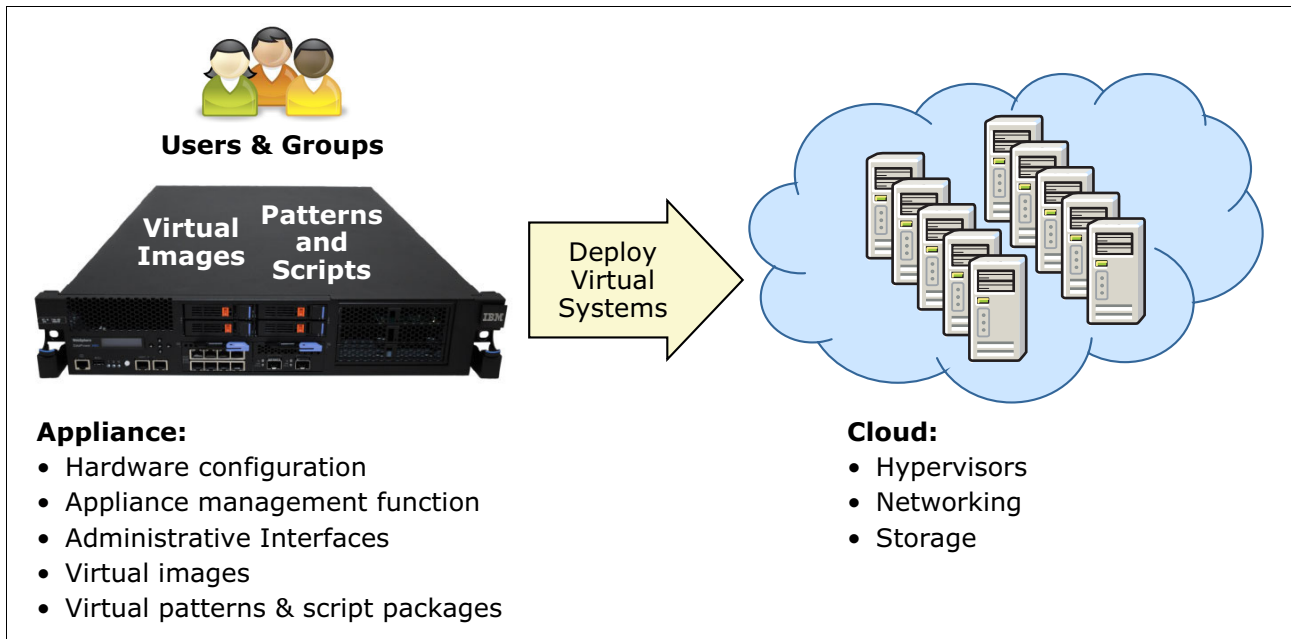


Figure 8-2 Platform components

These deployable virtual systems can be grouped as follows:

- ▶ Infrastructure as a Service (IaaS)
Delivers infrastructure elements, such as networks, memory, storage, and compute resources in a utility-based fashion.
- ▶ Platform as a Service (PaaS)
Provides an integrated platform consisting of infrastructure and middleware components to speed the development and delivery of applications.
- ▶ Software as a Service (SaaS)
Enables the consumption of applications and their data without the associated installation and maintenance costs of the supporting infrastructure.

Figure 8-3 on page 263 highlights the degree to which workload can be offloaded. Now organizations can consume IaaS and PaaS offerings and extend each using IBM Workload Deployer in order to build business applications quickly.

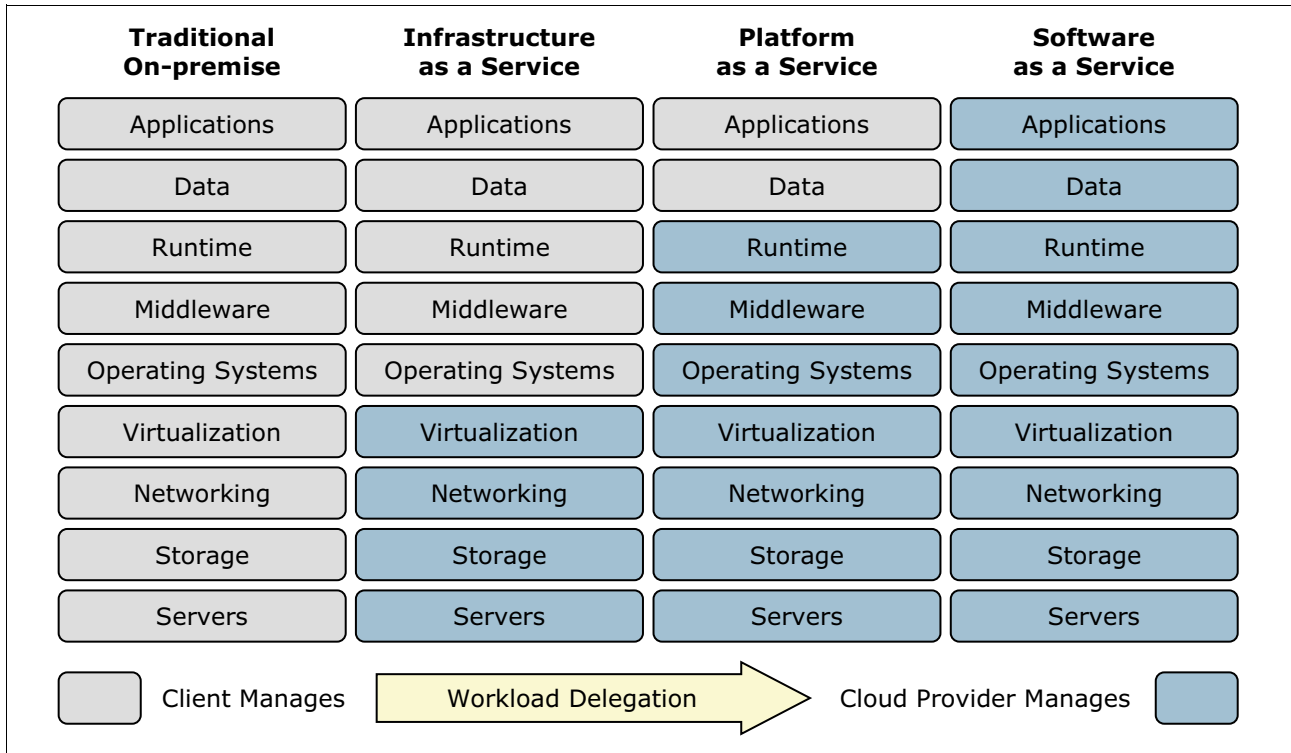


Figure 8-3 Cloud computing offerings

Conceptually all IT solutions consist of applications, platform, and infrastructure services. Figure 8-4 reflects the proportions of IT mass as it relates to cost, physical space requirements, maintenance, administration, management oversight, and obsolescence.

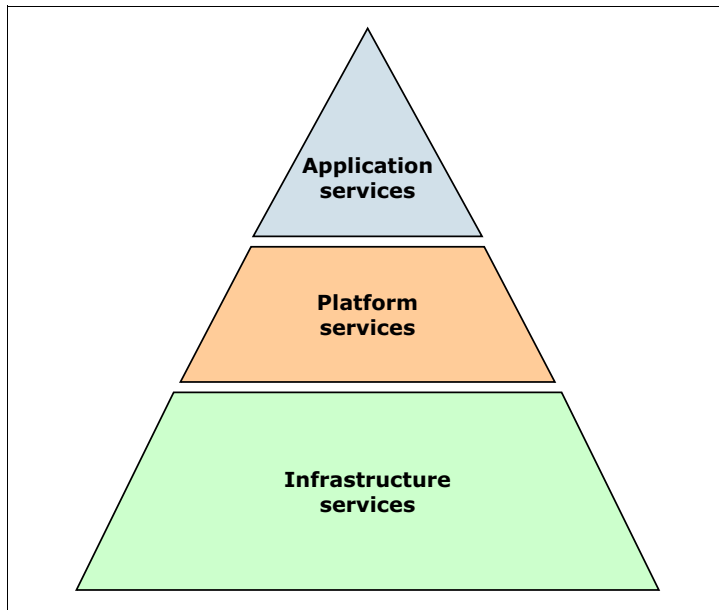


Figure 8-4 IT solution components

One of the biggest inhibitors in both development and test is the ability to acquire, deploy, configure, and host environments to perform development, prototyping, unit and full product testing. Cloud enables development and test teams to create and host such environments

rapidly and consistently. For development teams, they can now deliver continuous code integration and prototyping more effectively, while test teams can focus on testing the quality of a product or solution and worry less about enabling the testing. With IBM Workload Deployer, these teams can deploy any combination or multiples of these virtual images within existing resource constraints.

8.2 IBM Workload Deployer patterns

IBM Workload Deployer bases its ability to provision virtual systems, applications and databases to the cloud on the concept of patterns. Patterns are logical descriptions of both the physical and virtual assets that comprise a particular solution. The use of patterns allows an organization to develop an individual element or integrated solution once and then deploy it on demand as many times as needed thereafter.

IBM Workload Deployer provides two types of patterns to facilitate the rapid deployment and integration of private cloud capabilities: virtual system and virtual application patterns.

Virtual system patterns

Virtual system patterns represent repeatable topology definitions based on various middleware virtual images and runtime configurations. Virtual system patterns provide flexibility and control over the middleware topology to be deployed.

A virtual system pattern typically consists of an operating system and additional IBM software. An example of a virtual system pattern is one that contains Linux operating systems with WebSphere Application Server installed. The pattern might include a deployment manager, several custom nodes, an IBM DB2 database, along with an IBM HTTP Server.

Using a virtual system pattern gives you more control over the middleware topology but also requires that you configure the middleware. Script packages can be added to the pattern to automate the customization of the virtual system topology after it is up and running.

When a virtual system pattern is deployed, the appliance creates the topology, builds the relationships between the components (for example, federating the custom nodes to the deployment manager), and configures the middleware based on the script packages you provide. System administrators can log on to the deployed solution images to customize them further. This is shown in Figure 8-5 on page 265.

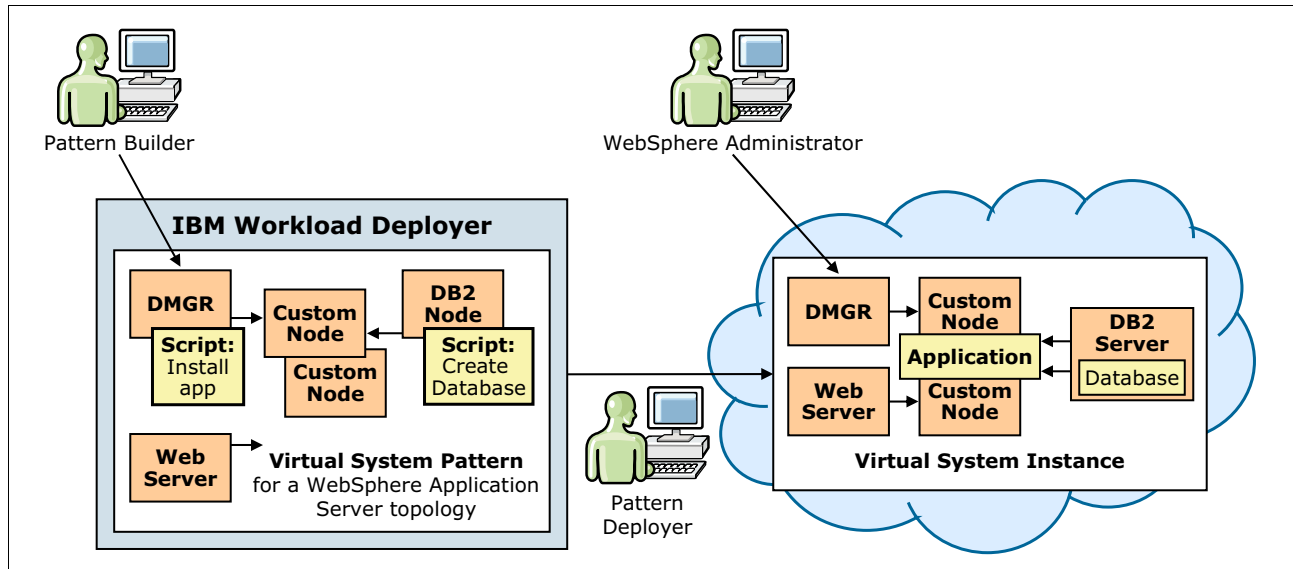


Figure 8-5 Deploying a virtual system pattern

Virtual Application patterns

While virtual system patterns focus on the topology, virtual application patterns—as the name implies—take an application-centric approach. With virtual application patterns, you describe an application and IBM Workload Deployer builds the appropriate infrastructure and deploys the application to it. IBM Workload Deployer includes a set of preloaded web application and database patterns. You may also create your own patterns, from scratch or by using any of the supplied patterns as a template.

Virtual application patterns are highly optimized and are constructed solely for supporting a single workload. This pattern requires the least amount of customization during deployment and provides the most direct method for obtaining a rapid return on investment. Virtual application patterns are application-centric, so you must provide the application files and describe the characteristics of how the application should be run and managed per given business policies. IBM Workload Deployer will generate the middleware topology to meet your requirements, as shown in Figure 8-6 on page 266.

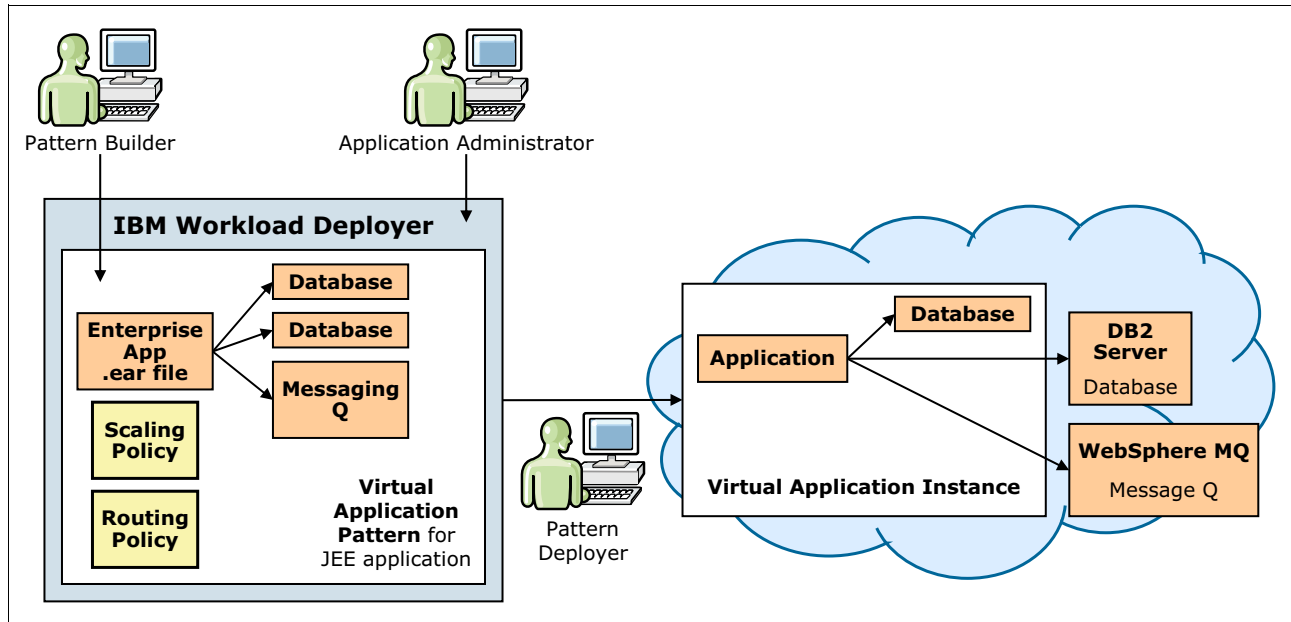


Figure 8-6 Provisioning a Virtual Application pattern

IBM Workload Deployer ships with pre-installed patterns of each type that represent varying degrees of automation and customization. All are optimized with the most appropriate configurations and settings for the solutions they support, because these pre-installed patterns are based on industry best practices.

Users can create their own patterns from any of the supplied patterns, or they are free to create new ones from scratch. Once a pattern has been stored on the appliance, it can be reused over and over to create multiple identical instances in a cloud as needed.

Pattern elements

Patterns can be constructed by integrating one or more elements. The integration activities can be as simple as standardizing the default location for software installation or as complex as automatic node federation in a WebSphere cell.

Figure 8-7 on page 267 offers a high-level view of the elements that can be used to construct patterns and the characteristics that define them.

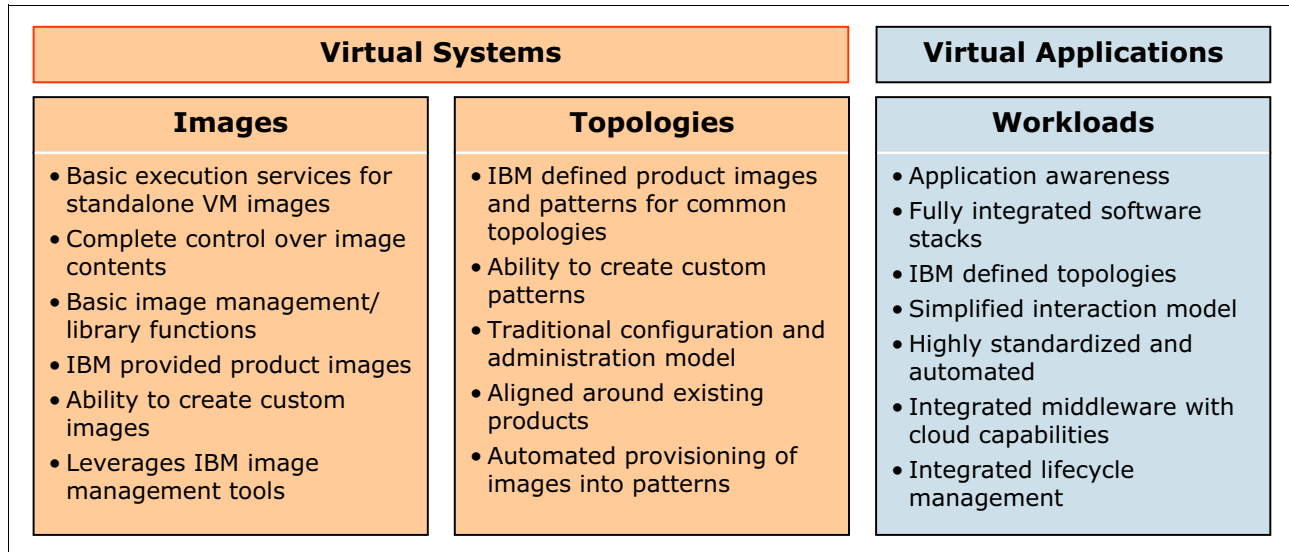


Figure 8-7 Pattern elements

Virtual images

Virtual images provide the operating system and product binary files needed to create a virtual system instance. IBM Workload Deployer supports a number of middleware Hypervisor Edition images, in the application infrastructure, business process management, connectivity, database, and portal arenas. These images are available for immediate use *as is* or can be customized to add extra functionality.

The appliance uses these images to create and deploy virtual machines to the cloud. The metadata describing the images follows Open Virtualization format (OVF), which is an industry standard specification for packaging and distributing virtual appliances that contain one or more virtual machines. Using OVF provides a standard mechanism to communicate virtual machine resource requirements to several hypervisors.

IBM Workload Deployer ships with a set of virtual images that includes:

- ▶ IBM WebSphere Application Server Hypervisor Edition virtual images for VMware ESX, IBM PowerVM, and IBM z/VM hypervisor technologies. This set of images also includes those images with the Intelligent Management Pack, which provides dynamic runtime capabilities similar to the capabilities of WebSphere Virtual Enterprise.
- ▶ IBM OS Image for Red Hat Linux Systems
- ▶ IBM operating system images for AIX
- ▶ WebSphere MQ Hypervisor Edition virtual machine images
- ▶ WebSphere Message Broker Hypervisor Edition virtual machine images
- ▶ DB2 Enterprise and Express Hypervisor Edition virtual images
- ▶ WebSphere Portal Hypervisor Edition virtual images

Users may extend this list by cloning an existing image, deploying it to the cloud, updating it, and then recapturing the “new” image.

In addition, users can create new images based on their operating systems using the IBM Image Construction and Composition Tool. This tool is described in 8.5.1, “IBM Image Construction and Composition Tool” on page 274.

Topologies

Virtual system patterns provide the topology for the virtual system to be provisioned to the cloud. The topology is created from the parts available in the images, plus optional customization scripts that provide configuration after deployment, like configuring WebSphere Application Server for an application workload and the installation of the application. The ability to integrate standard aspects of high availability and fault tolerance is contained within the topology.

Workloads

Users can achieve significant integration with middleware components and infrastructure resources, optimizing all for a particular type of application workload. Little knowledge of these underlying components is required before deploying and using the solution. The appliance can create or remove additional resources as required by the virtual application.

Virtual application patterns are deployed using IBM defined topologies, which are not exposed to the IBM Workload Deployer administrator. Rather, the topology is based on the application components defined by the virtual application pattern, and it gets adjusted in the runtime to accommodate the workload.

Administrative access to the middleware infrastructure running the application workload is obtained through a simplified interface on IBM Workload Deployer and not, for example, through an administrative console.

IBM Workload Deployer provides much of the decision-making actions related to deploying and running the application. However, policies can be configured by the pattern builder that define routing policies, policies for scaling workload, logging policies to specify the level of logging detail, and JVM policies. These policies allow users to configure the minimum and maximum heap sizes, JVM properties, verbose garbage collection, and other JVM-related settings.

8.2.1 Cloud resources

The cloud that virtual systems are deployed to consists of a set of hypervisors that the client provides. They need to configure the hypervisors to IBM Workload Deployer, then create pools of IP addresses to assign to provisioned systems on those hypervisors. Each hypervisor belongs to a cloud group. When a virtual system or virtual application is deployed, it is deployed to a cloud group. IBM Workload Deployer selects one or more hypervisors in the cloud group and assigns IP addresses to the provisioned systems from the IP group for the hypervisor. An alternative to having a static set of IP addresses is to use an environment profile to specify the IP address that is to be assigned by the pattern deployer.

Figure 8-8 on page 269 highlights the deployment process.

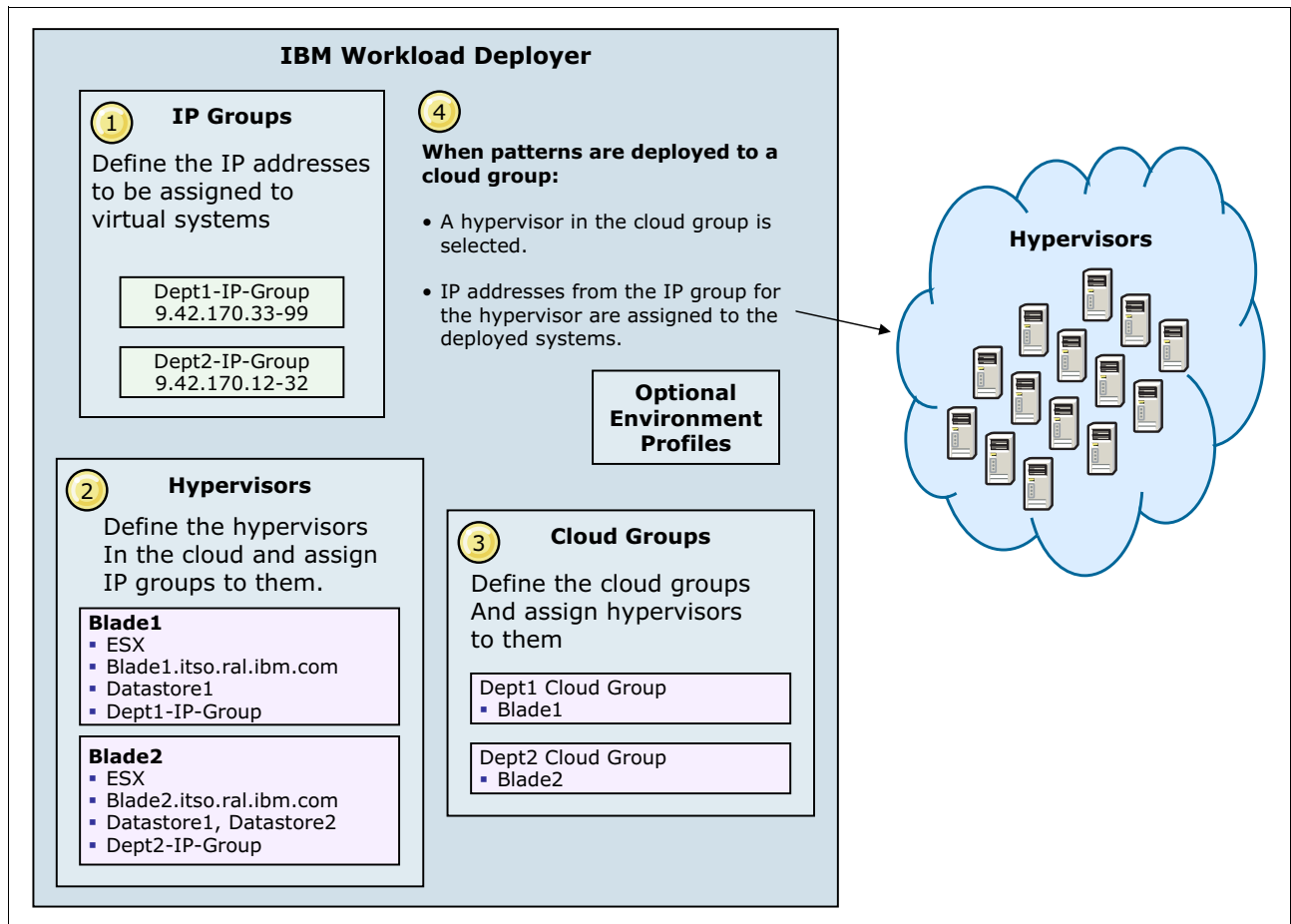


Figure 8-8 Deployment process

Hypervisors

A hypervisor is a software virtualization program that provides a layer of abstraction between operating systems and physical resources on a machine. This abstraction layer allows multiple operating systems and application stacks to run on a single physical entity and share resources, thus providing higher levels of resource utilization.

To set up the cloud, a system administrator defines the location and login credentials for the hypervisors. These hypervisors host the virtual systems that IBM Workload Deployer deploys. It automatically detects the storage that is associated with the hypervisors and manages the placing of the middleware virtual systems across the required set of hypervisors.

IBM Workload Deployer supports the following hypervisors:

- ▶ VMware ESX
- ▶ IBM PowerVM
- ▶ IBM z/VM

IP groups

Pools of IP addresses, known as *IP groups*, are configured for use by the deployed virtual images. A system administrator defines this pool of IP addresses, and when new virtual images are created, the appliance takes care of assigning each machine a unique value. IP addresses can then be added to and removed from the preconfigured pool as needed to accommodate demand.

Cloud groups

A cloud group is a collection of related hypervisors. When deploying patterns to create virtual systems, you use a cloud group as the deployment target. One or more hypervisors of the same type make up a cloud group, so you can group all of your ESX hypervisors together in one cloud group and all of your high-end PowerVM hypervisors together in another. A system administrator can manage resource allocation thresholds for the group, such as processor or memory usage, as well as verify the runtime status of any configured hypervisors.

In addition to managing cloud groups directly with IBM Workload Deployer, it can also be integrated with VMware vCenter to further streamline the deployment of ESX hypervisors. vCenter manages their deployment, while IBM Workload Deployer manages their placement.

Unsupported features: Both the Storage VMotion and Distributed Resource Scheduling vCenter features are not supported by IBM Workload Deployer at present and need to be disabled when configuring vCenter.

Environment profiles

Environment profiles group related deployment configurations, such as virtual machine names, IP address assignment, and cloud groups. Deploying patterns with environment profiles provide deployments across tiers from a single pattern.

In IBM Workload Deployer, environment profiles provide the functionality to:

- ▶ Define the operational environments, such as development, test, or quality assurance.
- ▶ Define virtual machine naming conventions in the operational environment.
- ▶ Specify whether the IP group or a pattern deployer provides the IP address on the deployment. Segment the clouds, and IP groups within the clouds, to specific environments.
- ▶ Assign aliases to the cloud resources, such as clouds and IP groups.
- ▶ Assign sections within the clouds to specific users or groups.
- ▶ Define limitations on the number of virtual processors, virtual memory, and storage.

With environment profiles, you can group multiple clouds to be used in the deployment, or they can deploy a pattern to multiple cloud groups of the same hypervisor type. However, you cannot deploy a single pattern to a z/VM cloud group and to a PowerVM cloud group. Environment profiles are platform-specific, so IBM Workload Deployer filters out the appropriate clouds.

8.3 Administrative interfaces

There are three ways to interact with IBM Workload Deployer:

- ▶ Web-based user interface (UI)
- ▶ Command-line interface (CLI)
- ▶ Representational State Transfer (REST) API

8.3.1 Web-based user interface

The primary administrative access to the IBM Workload Deployer appliance is through the web-based user interface (Figure 8-9 on page 271). This management console is enabled when the appliance is first initialized through the serial console.



Figure 8-9 IBM Workload Deployer user interface

The Welcome window provides wizards to configure, in a step-by-step approach, the core functionality of IBM Workload Deployer. When you click on the tab headers, you can access drop-down menus that allow them to accomplish related configuration tasks in a more granular way. For example, the cloud management options for the hypervisors, cloud and IP groups, can be found under the Cloud tab.

8.3.2 Command-line interface

The IBM Workload Deployer command-line interface (CLI) provides a scripting environment based on Jython, which is the Java-based implementation of Python. In addition to commands that are specific to IBM Workload Deployer, you can issue Python commands at the command prompt.

The CLI can be downloaded from the UI to either a Windows or Linux system, and it enables you to manage your IBM Workload Deployer appliance remotely. This interface communicates with the appliance over an HTTPS session, and it can be run in both interactive and batch modes.

8.3.3 Representational State Transfer API

The IBM Workload Deployer appliance exposes a subset of its function using a Representational State Transfer (REST) API. This API is available on the same IP address or host name used to access both the UI and CLI.

The REST API provides a means to interact with the appliance that is both language and programming model neutral. When using the REST API, a system administrator interacts with the appliance resources (such as hypervisors, patterns, script packages, and so on) just by using well-defined HTTP addresses and associated HTTP verbs, such as GET, POST, PUT, and DELETE.

Unlike the UI, the REST API is only supported over the HTTPS protocol, because the appliance uses a self-signed certificate for its SSL sessions. This same certificate is used for all three interface sessions. A system administrator needs to configure the HTTPS client to either accept or ignore this certificate during the SSL handshake and allow users to set the HTTP headers for each request.

Finally, the REST API supports only the sending and receiving of UTF-8 encoded data. A system administrator must also configure the HTTP client to encode and decode character data, including JSON data.

8.4 Appliance settings

After installation, the IBM Workload Deployer appliance settings can be performed through the administrative interfaces. This section provides a high-level overview of the administrative settings on the Workload Deployer appliance, covering networking, security, and basic appliance maintenance.

Networking

Networking settings can be configured for the appliance, including the Domain Name System (DNS), Network Time Protocol (NTP), and Simple Mail Transfer Protocol (SMTP) settings for the appliance. Although only a single Ethernet interface is required to be configured on the appliance for it to be functional, multiple Ethernet interfaces can be enabled. The most common reason for doing so is to add a level of redundancy to your environment. Another reason multiple Ethernet interfaces are used is to allow the appliance to separate the virtual machines network from the administrative one.

Security

IBM Workload Deployer is designed with key features that establish and manage trust across the cloud. In addition to ready-for-use security on the appliance, system administrators can also configure a Lightweight Directory Access Protocol (LDAP) to authenticate users with the Workload Deployer appliance.

Appliance maintenance

Using the backup and restore process, system administrators can capture a complete appliance environment at any point. They can then either restore this environment on the appliance from which it was taken or restore it on a different appliance.

Upgrades to the appliance are done using firmware updates. The firmware versions can be downloaded from the IBM Fix Central website. A firmware upgrade changes only the appliance application and does not affect the Hypervisor Edition virtual images on your appliance.

Power restart

The appliance can be restarted or powered down from the UI.

Users and groups

Users and user groups are configurable so that you can manage the level of access for each individual to your appliance.

User permissions are defined to determine which panels are viewable for each user and to determine a user's access to a particular object. Permissions provide the granularity to define the access and roles for each user. You specify access to patterns, virtual system instances, and catalog content at the object level.

The permissions assigned to users define the IBM Workload Deployer administrative tasks they can perform. In addition to determining which of the administrative pages are shown, the content of the Welcome page is dynamically generated by the appliance to display distinct content for users that are assigned dissimilar levels of access. For example, you can define the following role-based groups to control user access to resources on the appliance:

- ▶ Pattern deployers

This group has permission to deploy patterns. Typically, these users have less middleware administration expertise and probably want to deploy constructed and configured environments.

- ▶ Pattern authors and catalog managers

This group has permission to create patterns, upload script packages, and create custom images. These users are typically seasoned middleware administrators who can build and configure application environments. They map their existing configuration knowledge to the various customization approaches in IBM Workload Deployer.

- ▶ Cloud and appliance administrators

This group has permission to administer the cloud infrastructure and the appliance. These users are familiar with the configuration and administration of the hardware components in the cloud. In addition, they have the skills necessary to manage and maintain the appliance.

8.5 Tools for building custom assets

When the preloaded images, plug-ins and patterns do not meet all the requirements for your business workload, you can easily customize them using IBM Workload Deployer customization tools. The Pattern Editor is used to create and customize virtual system patterns, while the Virtual Application Builder allows you to customize virtual application patterns. These tools are accessible via the IBM Workload Deployer web user interface.

In addition, two new tools for creating custom content are available and can be downloaded from the IBM Workload Deployer UI Welcome window (see Figure 8-9 on page 271).

- ▶ The IBM Image Construction and Composition Tool (ICON) is used to create virtual images to be used in virtual system patterns.
- ▶ The Plug-in Development Kit (PDK) is used to create custom virtual application patterns.

8.5.1 IBM Image Construction and Composition Tool

IBM Workload Deployer includes IBM software product images with pre-installed middleware, such as WebSphere Application Server. Provisioning these preloaded images to your private cloud allows you to quickly build the virtual systems required to run your applications. Reality, however, is never that simple. It is likely that the images provided with the IBM Workload Deployer require enhancement before deployment.

For example, most organizations have corporate standards for security compliance across hardware and software platforms. The images deployed to the cloud must be updated to contain the software required to comply with your security policies. Another example is the requirement for the installation of agents used by monitoring software. Finally, consider the case where third-party products are required to be installed on the system. These changes could be made after deployment, but they would need to be performed after every deployment. As the number of virtual systems in the cloud grows, more time and effort is required to keep these systems in compliance.

A better solution is to customize the images before they are deployed. IBM Workload Deployer provides a “clone and extend” feature that allows users to deploy an image, update it, and then capture it as an image for future deployments. Although this feature has its advantages, you can take this concept one step further with IBM Image Construction and Composition Tool. Now you can prepare a software bundle that includes the information required to install and configure new software on a selected base image automatically at deployment time. These software bundles can be reused via the IBM Image Construction and Composition Tool to create additional customized images by mixing and matching bundles with base operating system images.

These customized images can be provisioned to a cloud using IBM Workload Deployer, IBM SmartCloud™ Enterprise, IBM SmartCloud Provisioning, or VMware ESX. A direct integration process is available between IBM Workload Deployer V3.1 and IBM Image Construction and Composition Tool V1.1.

Integration with IBM Workload Deployer

Integration between IBM Workload Deployer and the IBM Image Construction and Composition Tool takes place over a cloud provider connection defined in the tool interface. This link allows images in the IBM Workload Deployer image catalog to be viewable from within the tool.

Images from the appliance can be imported into the tool, effectively creating a representation of the image in the tool. The actual image stays on the appliance. With the image now in the tool, it can be extended by adding software bundles. When an image is extended, a new image description is generated using the imported image as a template, and the software bundles are added to this new image.

Next, the new image is synchronized via the tool, which creates a clone of the image and provisions that clone to the cloud. Once provisioned, the tool communicates with the cloud image instance and performs the installation and configuration tasks defined by the software bundles. At the completion of the synchronization process, your new virtual image can be tested. The last step is to use the tool to capture the extended image. The capture process cleans up the image to leave it in a state that is appropriate for redeployment. The captured image is defined in the tool and then added to the catalog of virtual images that resides in the appliance.

Figure 8-10 on page 275 illustrates this process.

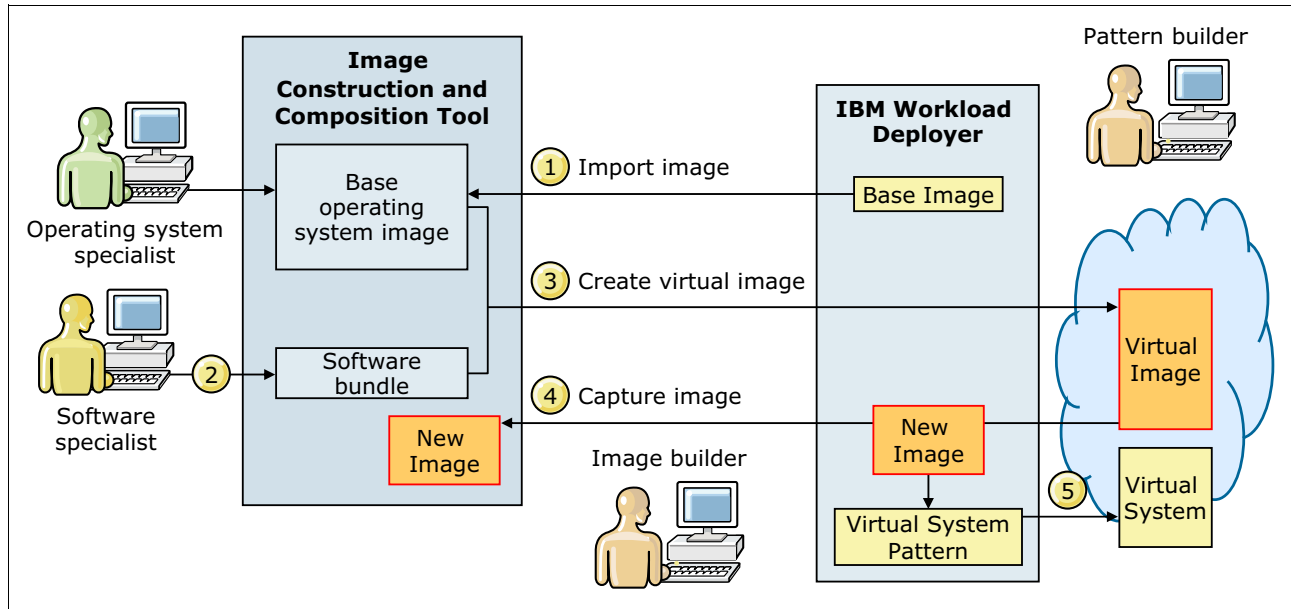


Figure 8-10 Customizing IBM Workload Deployer images

The steps shown in Figure 8-10 define the process for using IBM Image Construction and Composition Tool to build deployable images in IBM Workload Deployer, and they are described here:

1. An operating system specialist using IBM Image Construction and Composition Tool imports a base image directly from IBM Workload Deployer. The base image remains on IBM Workload Deployer and a description of the image is stored in IBM Image Construction and Composition Tool.
2. A software specialist defines the software bundles to be installed and configured on the base image. The software bundles contain installation and configuration parameters that define the middleware software to install to the virtual image:
 - Installation

The installation parameters define how the IBM Image Construction and Composition Tool installs software into the virtual image. The parameters are stored as a script that is run by the IBM Image Construction and Composition Tool after the image is built. The installation of software occurs one time, and the software then becomes a permanent part of the image.
 - Configuration

The configuration parameters contain a set of activation tasks that the IBM Image Construction and Composition Tool must perform to the installed software at deployment time.
3. The image builder creates a virtual image by specifying a base image to use and selecting the bundles to install. The image builder customizes the installation and configuration (deploy) parameters for the bundles as needed.

When this new image is saved, the IBM Image Construction and Composition Tool instructs IBM Workload Deployer to create a copy of the base image, install the bundles, and perform the configuration.

4. The image builder verifies the new system and then resets (cleans up) and captures the image.
5. The image is now available in the IBM Workload Deployer image catalog and can be included in virtual system patterns and deployed to the cloud.

Adding new images to IBM Workload Deployer

Figure 8-11 illustrates a second scenario for using IBM Image Construction and Composition Tool with IBM Workload Deployer to increase the flexibility building deployable virtual systems. In this example, the base image is not taken from an existing IBM Workload Deployer image. Instead, the base image is a virtual system that exists outside of the appliance domain.

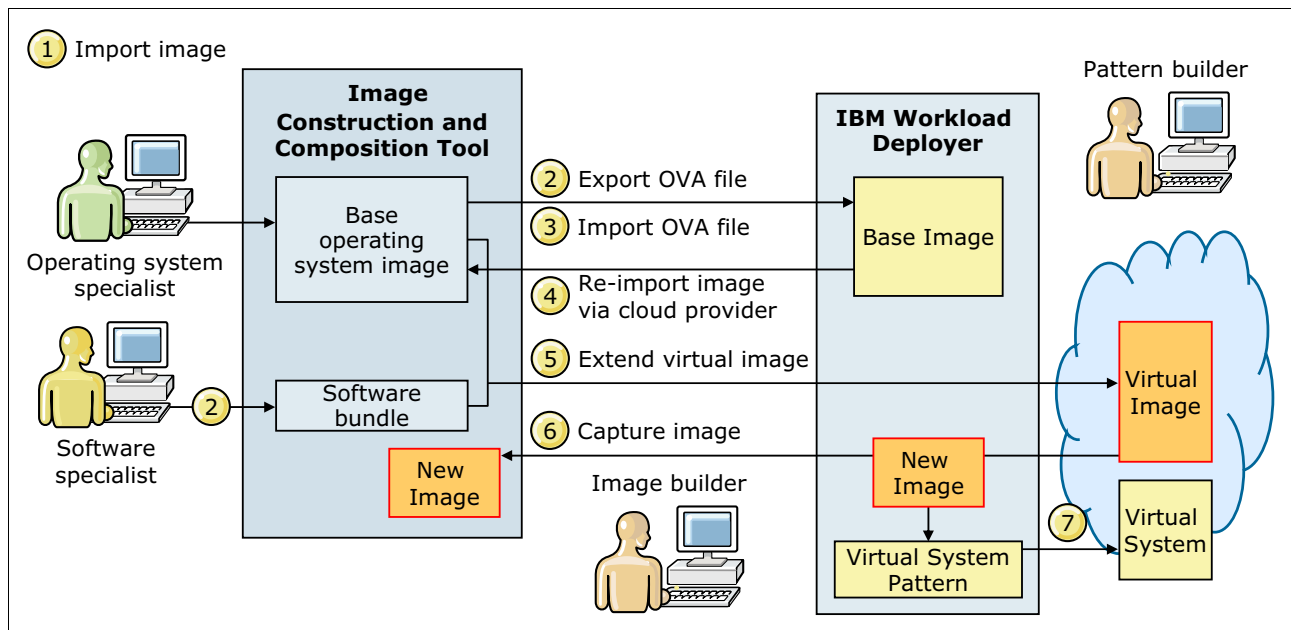


Figure 8-11 Building deployable images with the IBM Image Construction and Composition Tool

The steps shown in Figure 8-11 define the process to add new images to the IBM Workload Deployer image catalog using IBM Image Construction and Composition Tool, and they are explained here:

1. The operating system specialist defines the base operating system image to the IBM Image Construction and Composition Tool. In this case, the image is obtained from one of the following sources:
 - An existing virtual image (that is, an OVA file) with an operating system already installed is imported into IBM Image Construction and Composition Tool (Red Hat Enterprise Linux or SUSE Linux Enterprise Server only).
 - A running VMware virtual machine is captured into IBM Image Construction and Composition Tool.
2. The image builder then exports the new image for use with IBM Workload Deployer. The export process creates an OVA package that contains the metadata necessary for importing the image into, and deploying the image from, IBM Workload Deployer.
3. The IBM Workload Deployer administrator imports the image into the appliance.
4. The image builder re-imports the image into the IBM Image Construction and Composition Tool from the IBM Workload Deployer cloud provider. This action provides a new base image that can be extended.

5. A software specialist defines the software bundles to be installed and configured on the base image. The image builder creates a new virtual image by specifying the base image and selecting the bundles to install. The image builder customizes the installation and configuration (deploy) parameters for the bundles as needed. When this new image is saved, the IBM Image Construction and Composition Tool starts a copy of the base image, installs the bundles, and performs the configuration.
6. The image builder verifies the new system, and then captures the image.
7. The image is now available on your appliance for inclusion in virtual system patterns.

8.5.2 Plug-in Development Kit

Virtual application patterns take a decidedly application-centric approach for building, deploying and managing middleware application environments in a cloud. By supplying your application and specifying its functional and nonfunctional requirements, IBM Workload Deployer transforms your input into an installed, configured and integrated middleware application environment as shown in Figure 8-12.

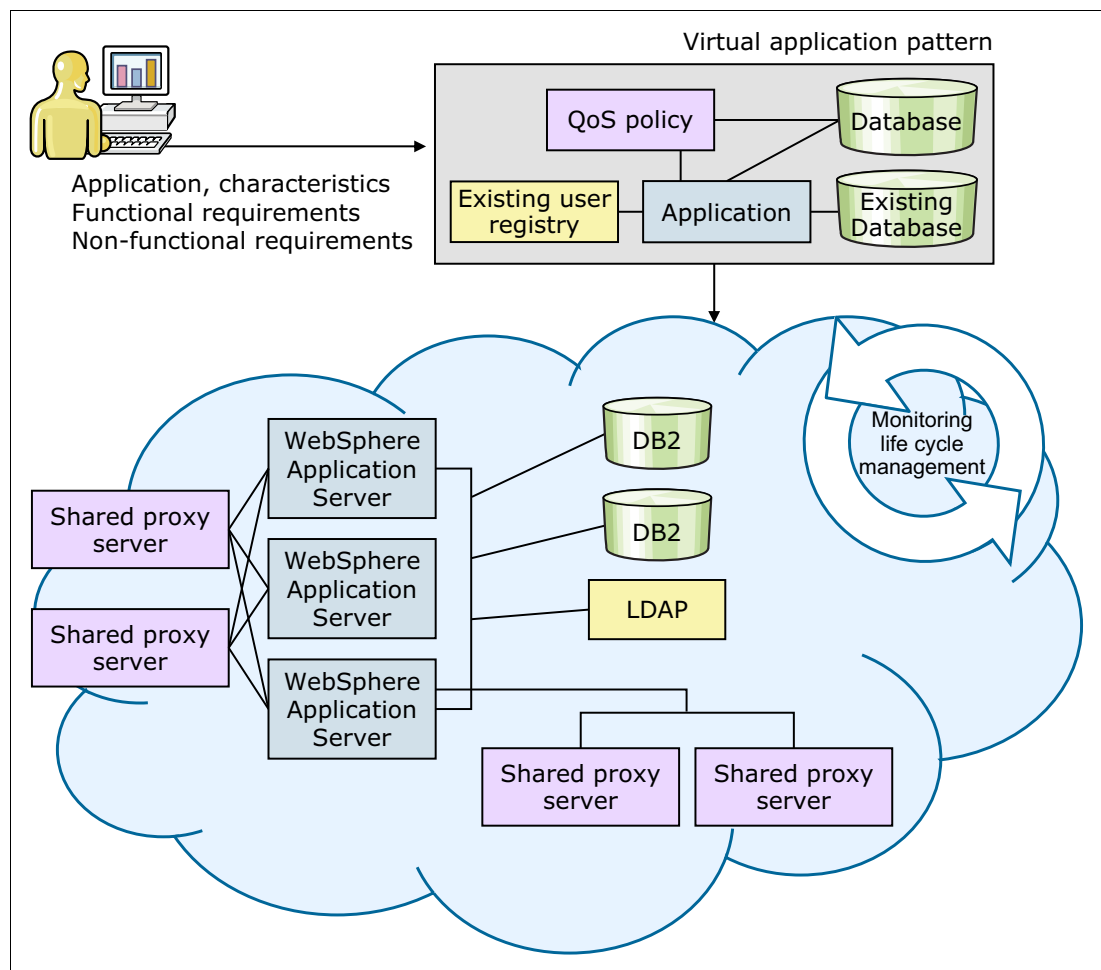


Figure 8-12 IBM Workload Deployer virtual application approach

The appliance frees those responsible for deploying virtual application patterns from having to be an expert in installing, configuring, or even integrating the middleware and applications, because the pattern encapsulates all that knowledge via plug-ins. The plug-ins for each pattern type define components, links, and policies and their functionality in the virtual application pattern, as highlighted in Figure 8-13.

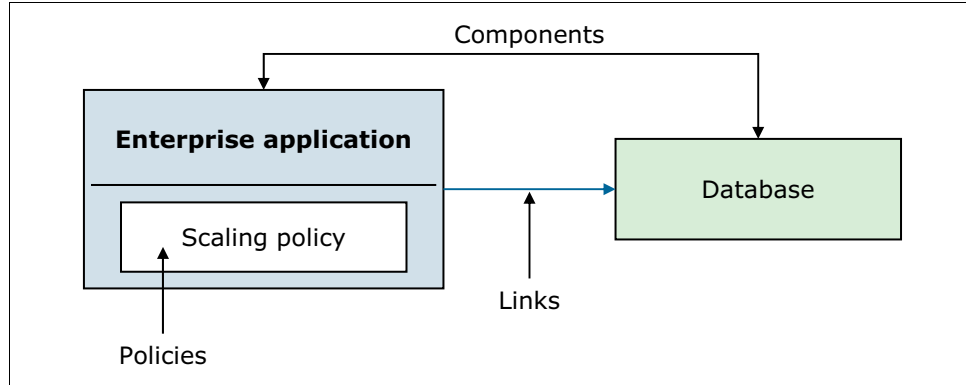


Figure 8-13 IBM Workload Deployer example virtual application pattern

In a virtual application pattern:

- ▶ Components represent functional profiles for a workload, such as enterprise application and database.
- ▶ Links express a connection point between components in a virtual application pattern. In the pattern shown in Figure 8-13, a link specifies that the Enterprise Application has a dependency on the Database component in the pattern.
- ▶ Policies like the Scaling policy shown in Figure 8-13 allow you to specify functional and nonfunctional requirements for your application environment.

With the IBM Workload Plugin Development Kit, you can create custom plug-ins, such as components, policies, and links, to extend an existing pattern or create a new pattern type.

8.6 Deploying virtual patterns to the cloud

To build a custom private cloud with IBM Workload Deployer:

1. Identify the hardware, hypervisors, and networking for the cloud.
2. Select and customize the virtual images.
3. Add script packages to customize the deployed middleware environment.
4. Use preinstalled or customized patterns to describe the middleware topology to be deployed, or you can build patterns from virtual images using drag-and-drop.
5. Deploy virtual systems to the cloud with the push of a button.

Figure 8-14 on page 279 shows the various components involved in building a private cloud, and the flow of operations.

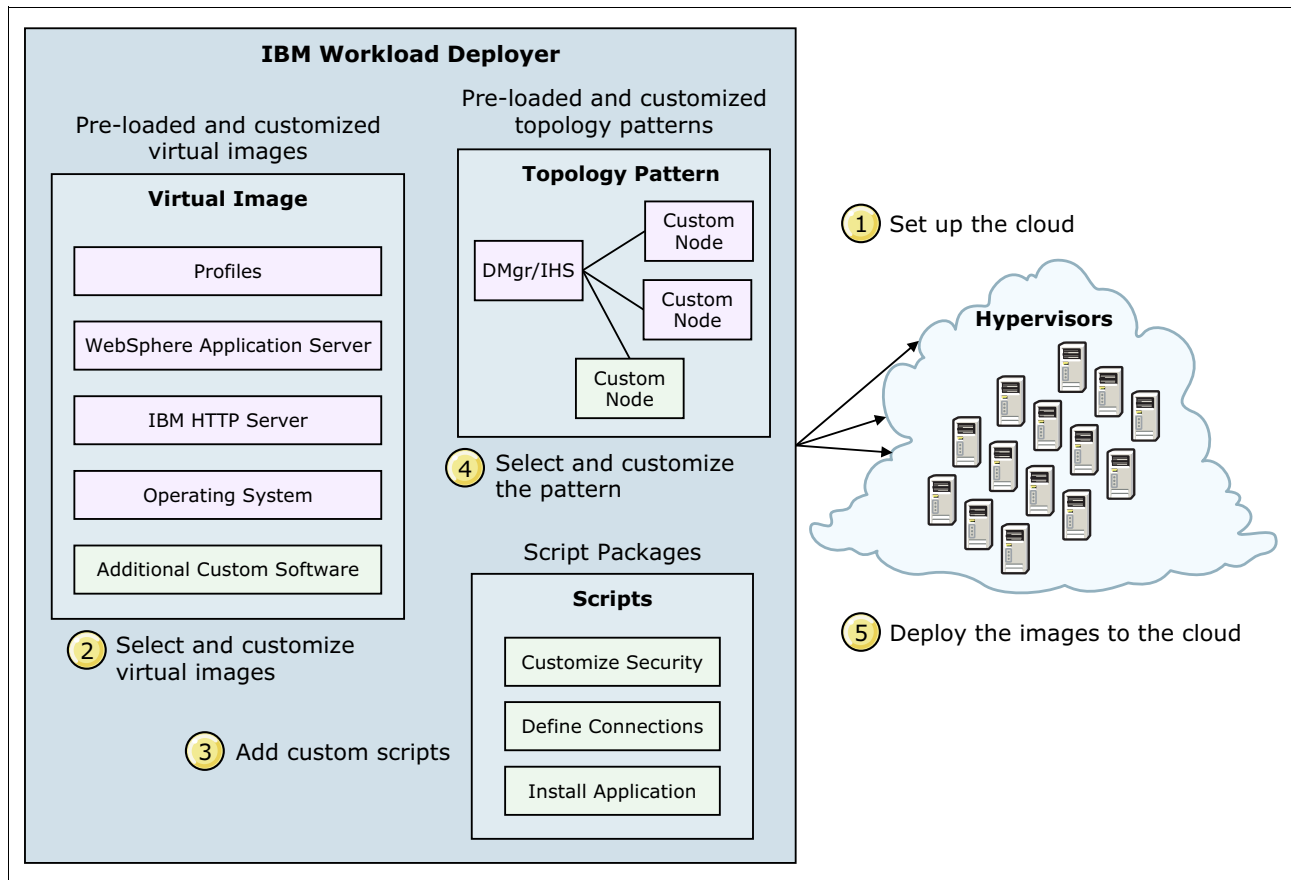


Figure 8-14 IBM Workload Deployer flow of operations

8.6.1 Deployment planning

It is anticipated that most application solutions share common resource requirements and similar expectations for service availability and performance. This is especially true for Internet or web-based applications that are commonly written in the Java programming language and share many of the same architectural characteristics. With the advent of a common programming language that is abstracted from the underlying systems, it becomes possible to aggressively pursue virtualization of the application platforms themselves. This enables the applications to be run on any application platform (hardware, operating system, and application server) so long as it adheres to a standard API.

Such a capability is what makes the construction of a private cloud attractive for IT departments. The introduction of application mobility makes it possible for organizations to refresh underlying physical hardware, apply system maintenance, or upgrade entire application platforms without incurring outages for the application itself.

Further, most applications undergoing continuous development will have workload characteristics change over time. Rapid introduction of new features or functions is common and it is not feasible for development teams to stop working while the infrastructure organization determines the resource impact of each change. The ultimate solution for this situation is to provide a mechanism for applications to scale above or below the hard resource allocations provided by physical servers or virtual guests. Private clouds provide this capability and can assist an organization in effectively managing the user experience.

As easy as the deployment process may seem with IBM Workload Deployer, some level of planning is required before any work begins to ensure that all the necessary prerequisites are identified and fulfilled. Proper planning up front helps avoid delays in creating your development, test or production environments.

Your planning efforts will be centered around two areas:

- ▶ System topology
- ▶ Managing cloud resources

System topology

Your system topology plan defines the products to be used and how they will form the solution. As you are defining the solution architecture, you will want to factor in requirements for service orchestration, networking, and security.

At a minimum, your planning should encompass:

- ▶ The use of discovered resources and standard configuration templates to reduce time and errors and improve communication of deployment requirements and the subsequent automation of provisioning tasks.
- ▶ The requirements for automating infrastructure provisioning, middleware configuration, and application installation to repeatedly set up standardized environments in the cloud, removing costly manual errors, and dramatically reducing provisioning times.
- ▶ Governance of shared application artifacts, standard templates, deployment plans, and trace development artifacts to deployed instances in order to support change management.

Managing cloud resources

Before starting the pattern development, configure your cloud resources. IBM Workload Deployer interfaces make this task easy.

As discussed in “Environment profiles” on page 270, an environment profile defines the configuration requirements for deploying a pattern. It provides an option to deploy a pattern to a specified cloud group. You can define profile information for the cloud, IP group, and IP address at a part level in an environment profile. You can select specific IP groups for each cloud and provide aliases to the cloud and IP groups to better describe the environment at deployment time. You can use the same pattern and deploy in different environments without changing the pattern.

With environment profiles, you can also group multiple clouds to be used in the deployment and you can deploy a pattern to multiple cloud groups of the same hypervisor type. You can work with the environment profiles via the IBM Workload Deployer CLI.

8.7 Applying a virtual application pattern to our scenario

We can use IBM Workload Deployer to deploy a virtual application pattern for the scenario used in this book. The scenario contains the products and components shown in Figure 8-15 on page 281.

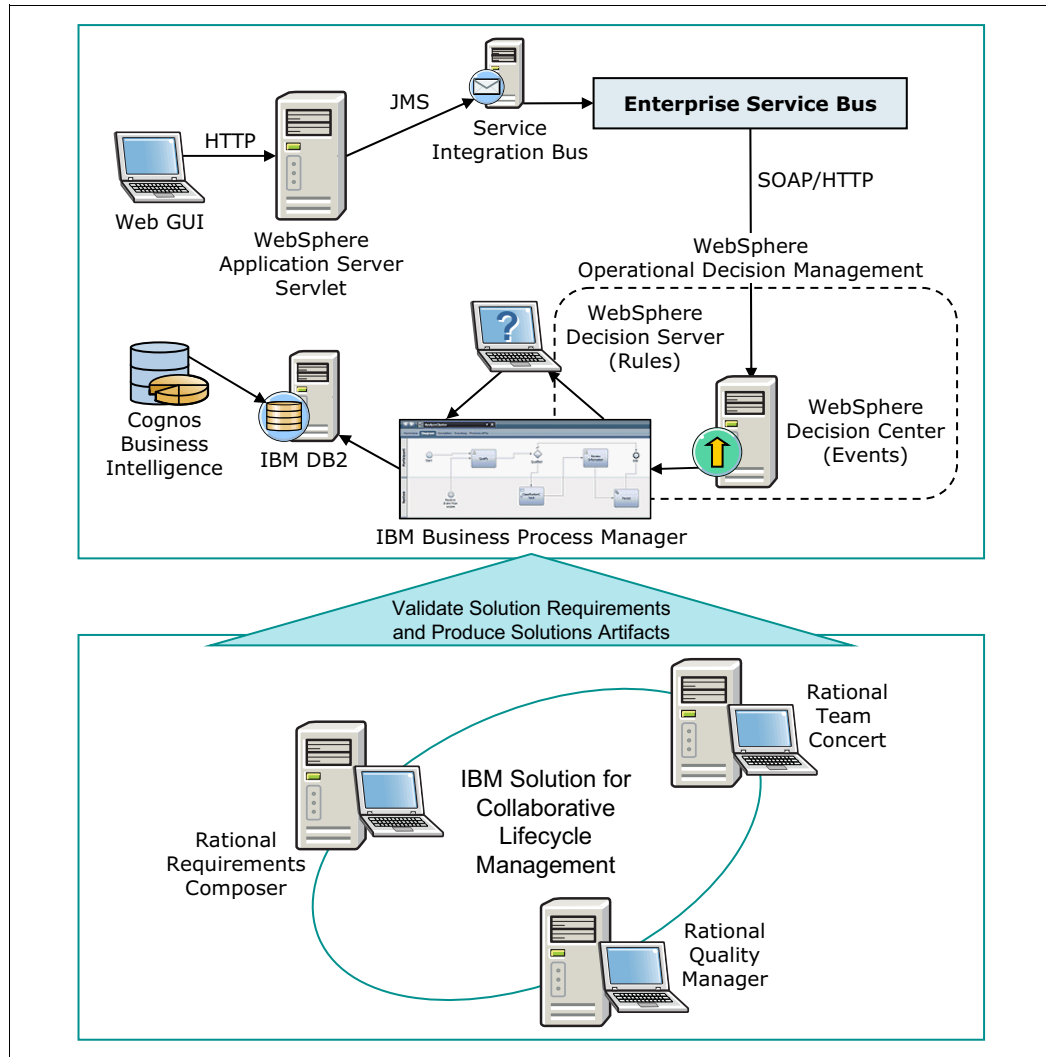


Figure 8-15 Our warranty application pattern

Since we have defined our application pattern in the preceding chapters, IBM Workload Deployer users can now create the requisite virtual images for each of the following products to complete a virtual application pattern for deployment via the appliance:

- ▶ IBM Business Process Manager Advanced V8
- ▶ IBM Cognos Business Intelligence V10.1.1
- ▶ IBM InfoSphere Data Architect V8.1.0.0
- ▶ IBM WebSphere Operational Decision Management V8
- ▶ IBM Solution for Collaborative Lifecycle Management V4.0

In a matter of days, you could have a working environment running in your private cloud, one that would allow you to refine this sample application further or test its capabilities. Using the integrated tools, you could ultimately develop a production-level solution, assuming the purchase of requisite software licenses, for deployment beyond your private cloud.

The core advantage provided by IBM Workload Deployer is the ability to create development, test and production environments quickly and with less overhead in a repeatable and consistent manner. Now you have a viable option to allow your resources to address the challenges of your business or industry in order to satisfy customer demand promptly.

Even better, your deployment options continue to grow. You can use the IBM Workload Plugin Development Kit to build virtual application patterns that can be deployed via IBM Workload Deployer, IBM PureApplication System, and IBM SmartCloud Application Workload Services. The IBM PureApplication System is one of the IBM PureSystems™ family of solutions, which are expert integrated systems that use built-in patterns of expertise based on years of IT experiences, client and partner engagements and best practices to automate many of the complex cloud-related development, deployment, and configuration tasks. In the following resources section, there are links to information sources for all these options, and more.

8.8 Additional information

In this chapter we provided a concise overview of the features and benefits for using IBM Workload Deployer. The following links let you obtain the “how to” details needed to harness this tool and others mentioned in the chapter.

IBM Workload Product Page

<http://www.ibm.com/software/webservers/workload-deployer/>

IBM Workload Deployer System Requirements

<http://www.ibm.com/software/webservers/workload-deployer/requirements/index.html>

IBM Workload Plugin Development Kit

<http://www.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-plugindekit>

IBM Workload Deployer Library

<http://www.ibm.com/software/webservers/workload-deployer/library/>

IBM Workload Deployer V3.1 Information Center

<http://pic.dhe.ibm.com/infocenter/worlodep/v3r1m0/index.jsp>

Redbooks

- ▶ *IBM Workload Deployer: Pattern-based Application and Middleware Deployments in a Private Cloud*, SG24-8011
- ▶ *Virtualization with IBM Workload Deployer: Designing and Deploying Virtual Systems*, SG24-7967

IBM Education Assistant Training

IBM Workload Deployer V3.1

<http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.iwd/iwd/IWD31.html>

developerWorks IBM Workload Deployer Group

<https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuiid=3d0d700d-6109-4375-88bb-9e628fc2a6ec>

IBM PureSystems

<http://www.ibm.com/ibm/puresystems/us/en/index.html>

developerWorks IBM PureSystems articles

- ▶ Overview: <http://www.ibm.com/developerworks/expert/index.html>

- ▶ Explore: <http://www.ibm.com/developerworks/expert/explore.html>
- ▶ Develop: <http://www.ibm.com/developerworks/expert/develop.html>
- ▶ Try: <http://www.ibm.com/developerworks/expert/try.html>

IBM WebSphere Hypervisor Edition

<http://www.ibm.com/software/websphere/products/hypervisorfamily/>

IBM SmartCloud Enterprise

<http://www.ibm.com/services/us/en/cloud-enterprise/>

IBM SmartCloud

<http://www.ibm.com/cloud-computing/us/en/>



Additional material

This book refers to additional material that can be downloaded from the Internet as described in the following sections.

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 at:

<ftp://www.redbooks.ibm.com/redbooks/SG248065>

Alternatively, you can go to the IBM Redbooks website at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG248065.

Restoring the IBM Cognos Business Intelligence examples

The additional materials contain a Framework Manager model and Cognos BI deployment archive. This section describes how to restore these examples.

Restoring the Framework Manager model

1. Locate the directory where the web material .zip file was extracted.
2. Extract the model from the **Cognos → FM_Model → Warranty Tracking Program Model.zip** file.
3. In IBM Cognos Framework Manager, use the open file command and browse to the cpf file that was extracted.

Restoring the IBM Cognos Business Intelligence deployment archive

1. Locate the directory where the web material .zip file was extracted.
2. Copy the Warranty_arch.zip file from the **Cognos** → **Deployment_archive** directory.
3. Paste the Warranty_arch.zip file to your active Content Manager system under the Cognos BI server installation location/deployment directory.
4. Open IBM Cognos Connection as an administrator user.
5. Launch IBM Cognos Administration from the Launch menu or the Manage Content link if the default start is on the Welcome panel.
6. In the Configuration tab, click **Content Administration**.
7. In the toolbar, click **New Import**.
8. In the Deployment archive box, click **Warranty_arch**.
9. Click **Next**.
10. Type a unique name and, if you want, enter a description, select the folder where you want to save it, and then click **Next**.
11. In the “Specify the general options - New Import Wizard” page, select whether to include access permissions and references to external namespaces, who should own the entries after they are imported, and the level of detail to save in the deployment record.

Note: Our deployment archive does not contain any references to external namespaces, or access permissions. All content was created in an unsecured IBM Cognos Business Intelligence installation.

12. Click **Next**.
13. Review the summary information and click **Next**.
14. Choose to run the import now, click **Save and run once**, and click **Finish**.
15. Click **OK**.

You should now be able to see the deployed content under the Public Folder in IBM Cognos Connection.



Creating a messaging infrastructure and web user interface

In this appendix we describe how to define the messaging interface and web user interface for the warranty claims process scenario used throughout this book. These components can be considered a part of an existing basic infrastructure that an organization may have in order to process warranty claims online.

Creating the messaging infrastructure

In this section we configure the necessary JMS infrastructure for the web user interface servlet to put messages on to JMS queues and for the JMS mediation to receive them.

Defining a service integration bus queue

First we define a new service integration bus queue. Perform the following steps:

1. Log into the WebSphere Integrated Solution Console. From the left navigation, choose **Service Integration** → **Buses**.
2. Click the appropriate bus where the messaging artifacts will reside. We chose **SCA.APPLICATION.IBMBPMNode01Cell.Bus** (Figure B-1 on page 288).

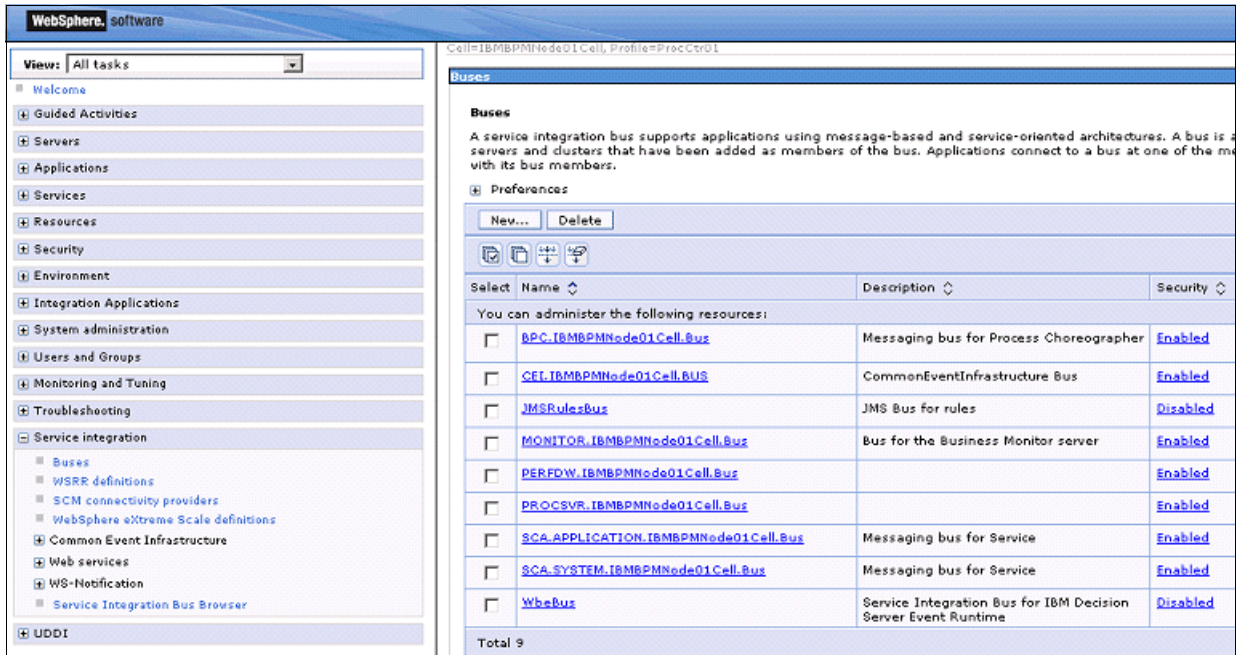


Figure B-1 Choose the appropriate bus in the Integrated Solution Console

3. Click **Destinations**. A list of existing destinations on this bus is displayed (Figure B-2).

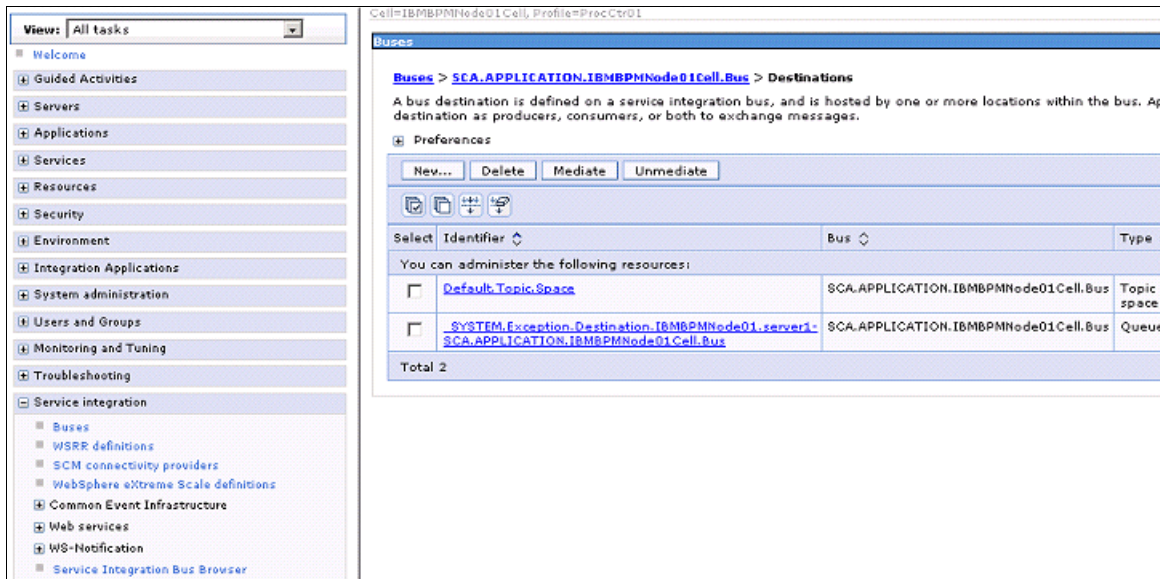


Figure B-2 Bus destinations

4. Click **New** to create a new destination. Specify the destination type as **Queue** and click **Next** (Figure B-3 on page 289).

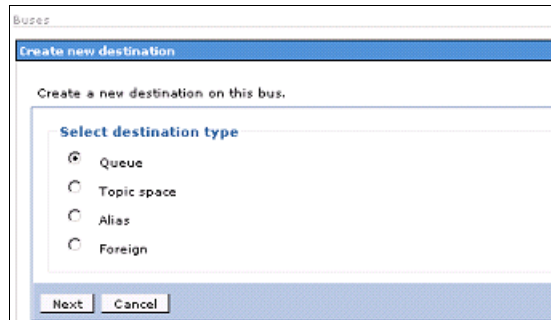


Figure B-3 Create a new destination (queue)

5. Enter an identifier name of EventInputQueue (Figure B-4) and click **Next**.

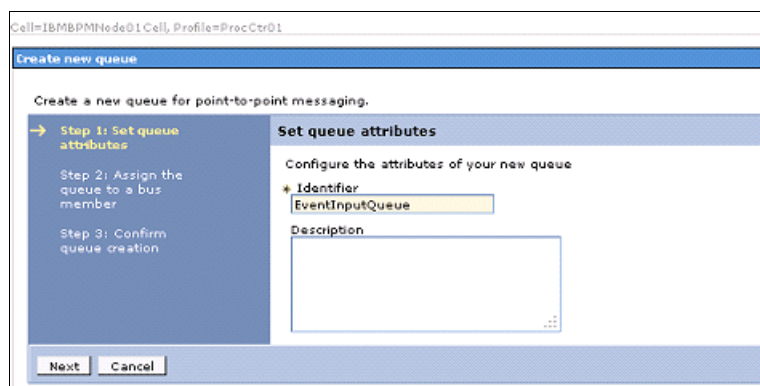


Figure B-4 Create a new queue

6. Leave the default bus member name (Figure B-5) and click **Next**.

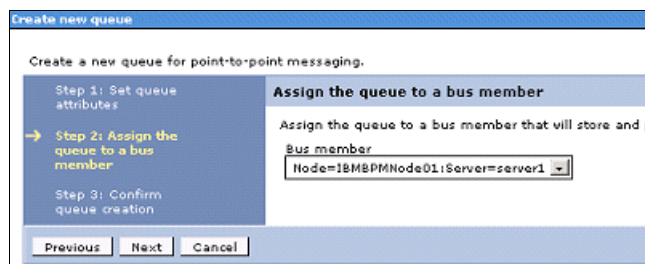


Figure B-5 Assign the queue to a bus member

7. Click **Save** to save the queue to the master configuration.

Defining JMS resources

With the queue destination defined, you can now define various JMS resources including a JMS queue, JMS queue connection factory, and JMS activation specification. Complete the following steps:

1. In the WebSphere Integrated Solution Console expand **Resources** and click **JMS providers** (Figure B-6 on page 290).

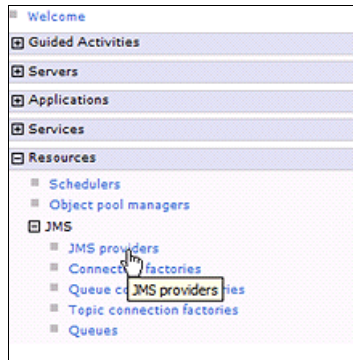


Figure B-6 JMS providers

2. Select **Default Messaging Provider** at the cell level (Figure B-7).

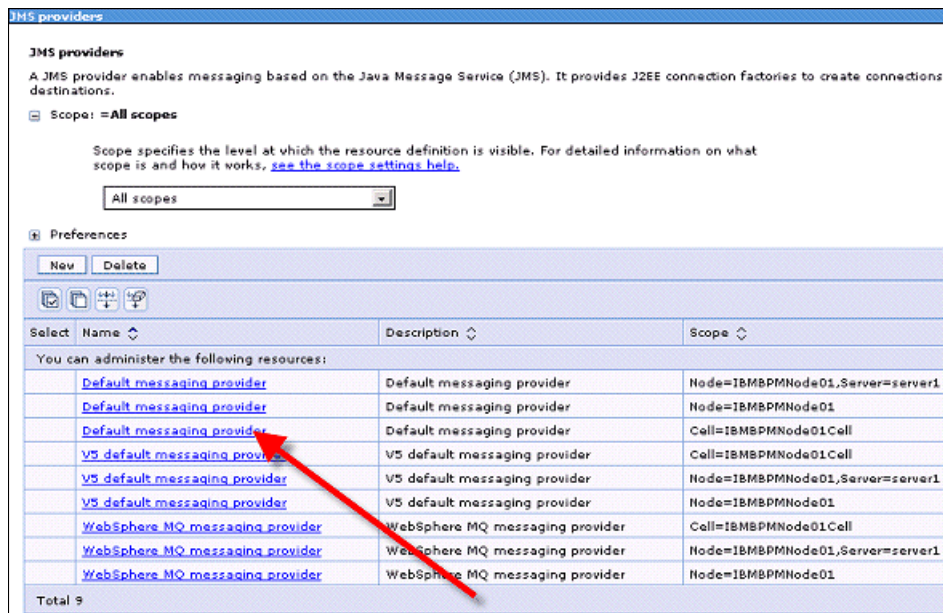


Figure B-7 Select the Default Messaging Provider at the cell level

3. Click **Queues** in the Additional Properties section, then click **New** to create a new JMS queue (Figure B-8). This new JMS queue will correspond to the queue destination you created in the service integration bus.

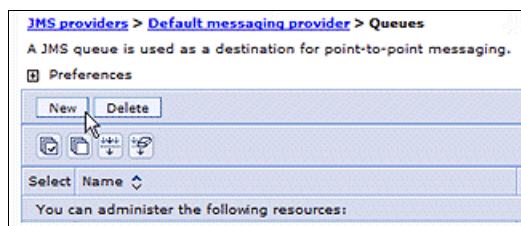


Figure B-8 Create a new JMS queue

4. In the Administration section, specify a Name of eventInputQueue and a JNDI name of jms/eventInputQueue. In the Connection section, select the bus you created your physical

queue on (**SCA.APPLICATION.IBMBPMNode01Cell.Bus**) and then select the queue you created earlier (**EventInputQueue**). See Figure B-9.

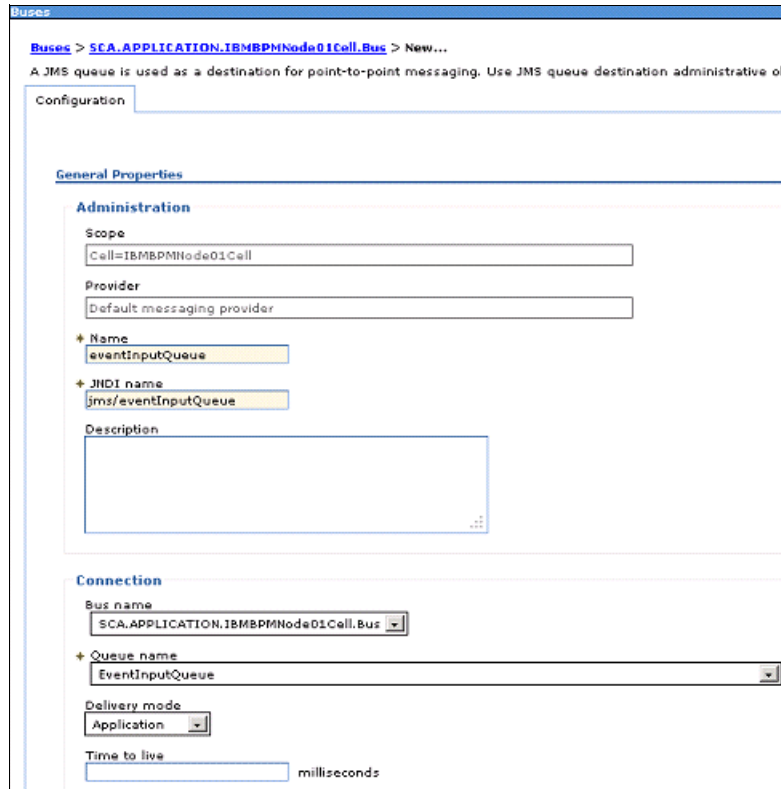


Figure B-9 Create a new JMS queue

5. Click **OK**.
6. Next create a JMS Queue Connection Factory that will be used to establish your JMS connection by the web user interface servlet. In the left navigation, click **Resources** → **JMS** → **Queue connection factories**. Select the scope of cell level as shown in Figure B-10.

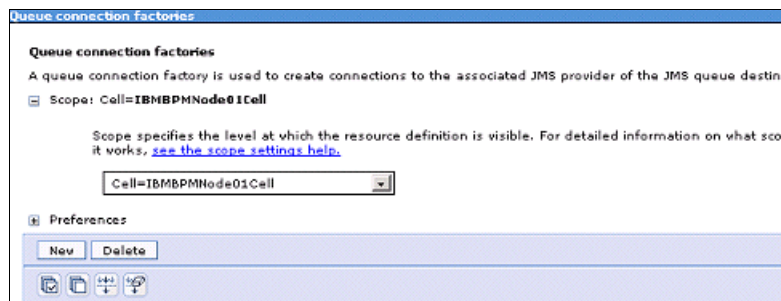


Figure B-10 Select cell level for queue connection factories

7. Click **New** to create a new queue connection factory.
8. Select **Default messaging provider** as the JMS resource provider for the queue connection factory.
9. In the Administration section specify the Name as eventQueueCF and the JNDI name as jms/eventQueueCF. In the Connection section, select the bus you created your physical

queue on (**SCA.APPLICATION.IBMBPMNode01Cell.Bus**) and click **OK**. See Figure B-11.

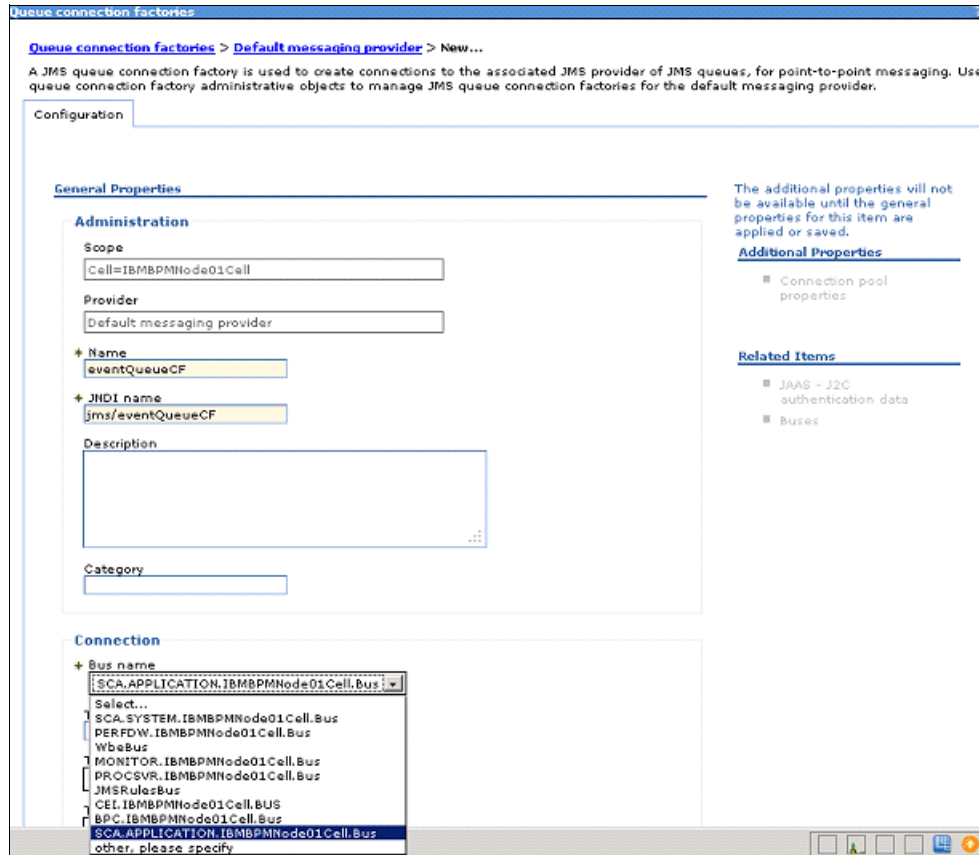


Figure B-11 Create a new queue connection factory

10. Save the settings for the queue connection factory.
11. Next create a JMS activation specification. In the navigation on the left panel, select **Resources** → **JMS** → **Activation specifications**. Click **New** to create a new activation specification (Figure B-12).

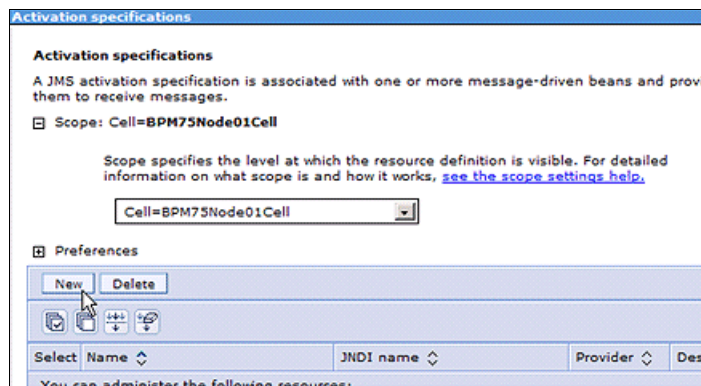


Figure B-12 Create a new activation specification

12. Select **Default messaging provider** as the JMS resource provider for the activation specification.

13. In the Administration section, specify a Name of actSpec and a JNDI name of jms/actSpec. In the Destination section, specify a name of jms/eventInputQueue and select the bus you created your physical queue on (**SCA.APPLICATION.IBMBPMNode01Cell.Bus**).

14. Click **OK** and save the settings for the activation specification.

Web user interface configuration

The web user interface configuration is performed with IBM Integration Designer. The following steps show how to create a simple web application to let you submit a serial number for a warranty item. This serial number will be placed on a JMS queue by a servlet that responds to an HTML form.

Creating a dynamic web project

Complete the following steps to create a dynamic web project:

1. Launch Integration Designer and choose **File** → **New** → **Dynamic Web Project**.
2. Enter a Project Name of RedbooksWebUI and select a Dynamic web module version of **2.4** (Figure B-13).

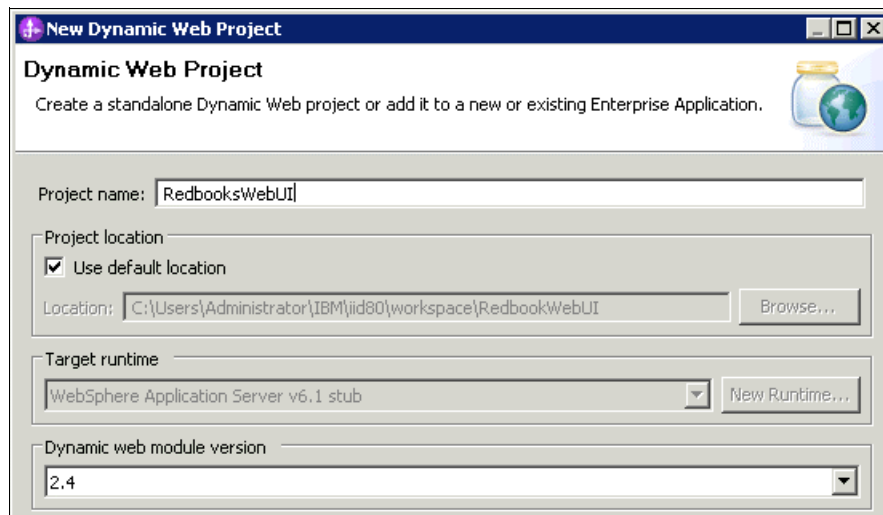


Figure B-13 New web project configuration dialog

3. In the EAR membership section, ensure that the **Add project to EAR** option is checked and click **New Project**.
4. Name the new project RedbooksWebUI_EAR.
5. Make sure that the target runtime is **IBM Process Server v8.0**.
6. Do not select any dependencies and click **Next** → **Finish**.
7. When asked if you want to open the associated perspective, click **Yes**.
8. You are now back to the Dynamic Web Project dialog. Click **Next**.
9. Click **Next** again.
10. When asked if you want to open the associated perspective, click **Yes**.

Creating a servlet

You just configured the basic project. Next you will create the servlet.

1. In Enterprise Explorer pane, expand the **RedbooksWebUI** project and expand the **RedbooksWebUI** descriptor that is inside the project.
2. Right-click the **Servlets** item in the RedbooksWebUI descriptor and select **New** → **Servlet**.
3. Enter the package name: `com.yourcompanyname.warranty`.
4. Enter the class name: `SenderServlet`.
5. Click **Next** → **Next** → **Finish**.
6. In the newly created `SenderServlet` servlet, replace the code with that in Example B-1 and save your changes.

Example: B-1 SenderServlet Java code

```
package com.yourcompanyname.warranty;

import java.io.IOException;
import java.io.PrintWriter;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.annotation.Resource;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.DeliveryMode;
import javax.jms.Destination;
import javax.jms.JMSException;
import javax.jms.MessageProducer;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class SenderServlet
 */
public class SenderServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Resource(name = "jms/eventQueueCF")
    private ConnectionFactory qcf;

    @Resource(name = "jms/eventInputQueue")
    private Destination myDestQueue;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public SenderServlet() {
```

```

        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        System.out.println("In doPost Method");
        PrintWriter out = response.getWriter();
        out.println("<html><head>");
        out.println("<title>Acknowledgement of input data</title>");
        out.println("</head></html></body>");
        String serialNumber = request.getParameter("serialNumber");
        out.println("<p>Serial Number: " + serialNumber + " received</p>");
        putMsgOnQueue(serialNumber);
        out.println("<p>Message place on queue</p>");
    }

    private void putMsgOnQueue(String serialNumber) {
        Connection con = null;
        try {
            System.out.println("Creating connection to Queue Connection Factory");
            con = qcf.createConnection();
            System.out.println("Connection established to Queue Connection
Factory");
            Session session = con
                .createSession(false, Session.AUTO_ACKNOWLEDGE);
            System.out.println("Connection Session Created");
            MessageProducer sender = session.createProducer(myDestQueue);
            System.out.println("Message Producer created");
            System.out.println("Creating TextMessage Object");
            System.out.println("Populating textmessage object");
            TextMessage textMessage = session.createTextMessage();
            DateFormat dateFormat = new SimpleDateFormat(
                "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'");
            Date date = new Date();
            String formattedDate = dateFormat.format(date);
            String message = "<?xml version='1.0' "
encoding="UTF-8'?"><APPMESSAGE>
                + "<SERIAL_ID>"
                + serialNumber
                + "</SERIAL_ID>"
                + "<TIMESTAMP>"

```


Configuring the servlet to point to the JMS components

You have now completed the HTML page to call the servlet. The last step is to configure the servlet so that it points to the correct JMS components and can authenticate to the JMS connection factory.

1. In the **RedbooksWebUI** project, double-click the deployment descriptor **RedbooksWebUI**.
2. Next to References, click **Details**.
3. Click **Add**, select **Resource Reference** and click **Next**.
4. Enter the following values
 - a. Name: `jms/eventInputQueue`
 - b. Type: `javax.jms.Queue`
 - c. Authentication: Container
5. Click **Finish**.
6. Click **Add**, select **Resource Reference**, and click **Next**.
7. Enter the following values
 - a. Name: `jms/eventQueueCF`
 - b. Type: `javax.jms.ConnectionFactory`
 - c. Authentication: Container
8. With **jms/eventQueueCF** still selected in References, go to WebSphere Bindings and select **JAAS login configuration** and choose **Use default method**.
9. Enter the Authentication Alias of `SCA_Auth_Alias`.

The servlet configuration is now complete and the final deployment descriptor configuration should look like Figure B-14.

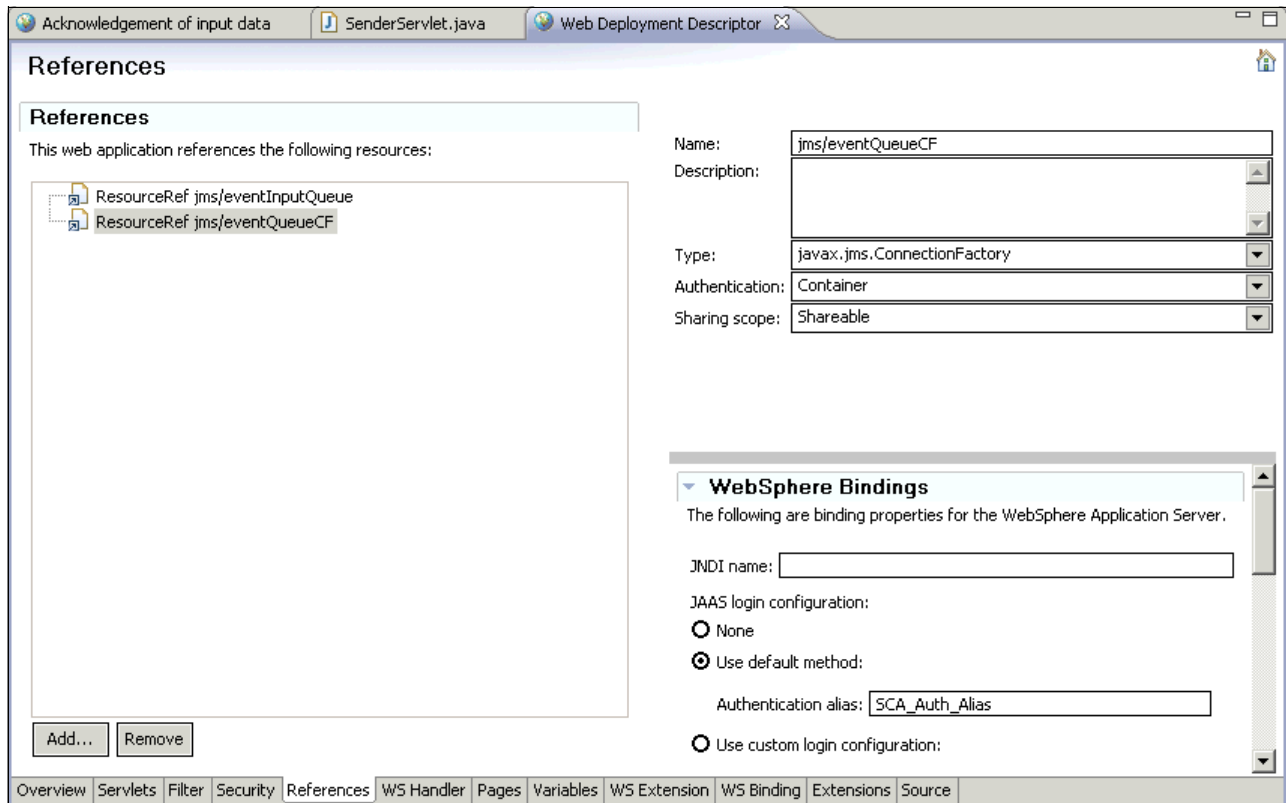


Figure B-14 Servlet deployment descriptor after configuration

As you have selected WebSphere Process Server v8.0 as your deployment target, you will see the following message in the Problems pane: The annotation Resource will not be recognized by WebSphere Application Server v7.0 runtimes or above. The manifest.mf needs to be changed so that the annotations will be recognized. This can be performed simply by right-clicking this message, selecting **Quick Fix**, then clicking **Finish**.

Running the web user interface application

In order to run the web application:

1. Right-click **test.html** and select **Run As** → **Run on Server**.
2. Select **IBM Process Server v8.0**.
3. Add **RedbookWebUI_EAR** to the projects that must be deployed.
4. Click **Finish**.

The web application will now execute. You can now enter a serial number that will be placed on the JMS queue which was configured previously. If you have created the Event Rules project, this will trigger an event and then, if multiple submits are performed in quick succession for the same serial number, it should also trigger the business process to handle multiple warranty claims.

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

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *Combining Business Process Management and Enterprise Architecture for Better Business Outcomes*, SG24-7947
- ▶ *IBM Workload Deployer: Pattern-based Application and Middleware Deployments in a Private Cloud*, SG24-8011
- ▶ *Virtualization with IBM Workload Deployer: Designing and Deploying Virtual Systems*, SG24-7967
- ▶ Making Better Decisions Using IBM WebSphere Operational Decision Management at:
<http://www.redbooks.ibm.com/abstracts/redp4836.html?Open>

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Redbooks

Implementing an Advanced Application Using Processes, Rules, Events, and Reports

(0.5" spine)
0.475" x 0.873"
250 <-> 459 pages



Implementing an Advanced Application Using Processes, Rules, Events, and Reports

**Defining processes
with Business
Process Manager and
WebSphere
Operational Decision
Management**

**Data modeling with
InfoSphere Data
Architect**

**Reporting with
Cognos Business
Intelligence**

In this IBM Redbooks publication we describe how to build an advanced business application from end to end. We use a fictional scenario to define the application, document the deployment methodology, and confirm the roles needed to support its development and deployment. Through step-by-step instructions you learn how to:

- ▶ Define the project lifecycle using IBM Solution for Collaborative Lifecycle Management
- ▶ Build a logical and physical data model in IBM InfoSphere Data Architect
- ▶ Confirm business rules and business events using IBM WebSphere Operational Decision Management
- ▶ Map a business process and mediation using IBM Business Process Manager
- ▶ Use IBM Cognos Business Intelligence to develop business insight

In addition, we articulate a testing strategy using IBM Rational Quality Manager and deployment options using IBM Workload Deployer.

Taken together, this book provides comprehensive guidance for building and testing a solution using core IBM Rational, Information Management, WebSphere, Cognos and Business Process Management software. It seeks to demystify the notion that developing and deploying advanced solutions is taxing.

This book will appeal to IT architects and specialists who seek straightforward guidance on how to build comprehensive solutions. They will be able to adapt these materials to kick-start their own end-to-end projects.

**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**

SG24-8065-00

ISBN 0738437387