Introducing IBM FileNet Business Process Manager
August 2008
First Edition (August 2008)

This edition applies to Version 4, Release 0 of IBM FileNet Business Process Manager (product number 5724-R76).

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

Notices ....................................................................................... ix
Trademarks ................................................................................. x

Preface ......................................................................................... xi
The team that wrote this book ................................................... xi
Become a published author ...................................................... xiii
Comments welcome ..................................................................... xiv

Chapter 1. Overview ................................................................. 1
1.1 Business process management .............................................. 2
   1.1.1 Transactional processes and content centric processes .......... 3
1.2 IBM FileNet Business Process Manager overview .................... 4
1.3 Process design with BPM ..................................................... 6
   1.3.1 Process definition .......................................................... 6
   1.3.2 Steps ........................................................................... 7
   1.3.3 Routes ......................................................................... 8
   1.3.4 Maps .......................................................................... 9
   1.3.5 Deadlines and timers ................................................... 9
   1.3.6 Content events ........................................................... 10
1.4 BPM application development options ................................... 10
   1.4.1 IBM FileNet Workplace (or Workplace XT) ....................... 10
   1.4.2 Business Process Framework (BPF) .............................. 11
   1.4.3 Electronic forms (eForms or Lotus Forms) ...................... 11
   1.4.4 Portal user interface through IBM FileNet P8 portlet .......... 12
   1.4.5 Web Application Toolkit (WAT) .................................... 12
   1.4.6 Process Engine API .................................................... 12
1.5 BPM integration to external systems ..................................... 12
   1.5.1 Component Integrator .................................................. 12
   1.5.2 Web services .............................................................. 13
   1.5.3 Business rules engines ................................................. 13
1.6 BPM process monitoring and analysis .................................... 13
   1.6.1 Process Simulator ....................................................... 14
   1.6.2 Process Analyzer ........................................................ 15
   1.6.3 IBM FileNet Business Activity Monitor ......................... 16
1.7 BPM applications and tools summary .................................... 17
1.8 The IBM FileNet P8 family of products ................................ 18
   1.8.1 IBM FileNet content products .................................... 18
   1.8.2 IBM FileNet process products .................................... 19
1.8.3 IBM FileNet compliance products ........................................ 20
1.8.4 IBM FileNet system management tools ............................... 21

Chapter 2. Solution example ...................................................... 23
2.1 Scenario background ......................................................... 24
2.2 Claim approval process ...................................................... 24
   2.2.1 Initiating a claim (steps 1 and 2) ..................................... 26
   2.2.2 Back office mailroom clerk task (step 3) ......................... 28
   2.2.3 Claim processing (steps 4 through 6) ............................. 29
   2.2.4 Financial controls (part of steps 5 through 6 and 7) ............ 30
   2.2.5 Workflow process maps for the claim approval process ....... 32
2.3 Summary of benefits .......................................................... 33

Chapter 3. System architecture ............................................... 35
3.1 IBM FileNet P8 core engines. ............................................. 36
3.2 Layered architecture .......................................................... 37
   3.2.1 Presentation tier ...................................................... 39
   3.2.2 Middleware tier ....................................................... 41
   3.2.3 Data tier ............................................................... 42
3.3 Process Engine (PE) applications and other supporting applications ... 43
   3.3.1 Process Simulator system architecture .......................... 44
   3.3.2 Process Analyzer system architecture ......................... 44
   3.3.3 Business Activity Monitor system architecture ............. 45
   3.3.4 Business Process Framework system architecture .......... 46
3.4 Service Oriented Architecture (SOA) .................................... 47
3.5 Process Engine logical view ................................................ 49

Chapter 4. Process design with BPM ........................................... 53
4.1 Basic concepts in process design ......................................... 55
   4.1.1 Tasks, participants, and roles .................................... 55
   4.1.2 Workflow properties (workflow groups and attachments) .... 57
   4.1.3 Steps ................................................................. 58
   4.1.4 Routing ............................................................... 58
   4.1.5 Milestones ........................................................... 60
   4.1.6 Deadlines ............................................................. 60
   4.1.7 Task reassignment .................................................. 61
   4.1.8 Process voting ....................................................... 61
4.2 BPM supported data types and system functions .................... 62
   4.2.1 Supported data types ............................................... 62
   4.2.2 System functions ................................................... 63
4.3  Inter-related processes .................................................. 64
4.4  Interaction with external systems ................................. 65

Chapter 5. Using Process Designer ............................................. 67
5.1  Process Designer user interface overview ....................... 68
   5.1.1  Map pane ............................................................. 70
   5.1.2  Properties pane .................................................... 71
   5.1.3  Step Palettes pane ................................................ 71
   5.1.4  Toolbar icons ...................................................... 75
5.2  Creating workflow definition overview ............................ 76
5.3  Setting up workflow properties ...................................... 77
5.4  Creating steps in a workflow map .................................... 88
   5.4.1  General step ....................................................... 91
   5.4.2  Submap step ....................................................... 101
   5.4.3  System step ....................................................... 105
   5.4.4  Component step .................................................. 108
   5.4.5  Launch step ...................................................... 110
   5.4.6  Start step .......................................................... 111
5.5  Creating routes with Process Designer .......................... 111
   5.5.1  Setting conditions for routes .................................. 112
5.6  Advanced topics .......................................................... 117
   5.6.1  Workflow inheritance ............................................. 117
   5.6.2  Default system submaps ........................................ 121
   5.6.3  Expression Builder .............................................. 122

Chapter 6. Implementing business processes: Case study ........ 127
6.1  Building business processes overview ......................... 128
6.2  Defining the basic process object model ....................... 129
6.3  Starting a new workflow process .................................. 132
   6.3.1  Defining workflow properties ................................ 134
   6.3.2  Saving the workflow process ................................ 142
6.4  Building the process workflow map ............................... 146
   6.4.1  Adding steps ..................................................... 146
   6.4.2  Adding routes ................................................... 149
   6.4.3  Setting up conditional routing ................................. 152
   6.4.4  Validating the workflow process map ....................... 157
6.5  Finishing the step properties definition ......................... 159
   6.5.1  Setting up a component step .................................. 169
6.6  Building a submap ....................................................... 172
   6.6.1  Creating the Claim Setup submap ............................ 172
   6.6.2  Setting up system steps ....................................... 181
   6.6.3  Linking the submap to the main map ....................... 184
   6.6.4  Checking in and transferring your workflow process map 186
Notices

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

- DB2®
- Domino®
- FileNet®
- IBM®
- Lotus®
- OmniFind™
- Redbooks®
- Redbooks (logo) ®
- WebSphere®
- Workplace™

The following terms are trademarks of other companies:

- Cognos, and the Cognos logo are trademarks or registered trademarks of Cognos Incorporated, an IBM Company, in the United States and/or other countries.

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

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

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

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

- Excel, Expression, Internet Explorer, Microsoft, Visio, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

This IBM® Redbooks® publication provides a basic introduction to IBM FileNet® Business Process Manager (BPM) V4.0. BPM enables organizations to create, modify, manage, simulate, and analyze content centric business processes. One key advantage of BPM is its ability to work with active content, which refers to the ability of content to trigger or affect business processes.

In this book, we cover the key elements that make up a business process, including tasks, participants, roles, steps, routing, and deadlines. We describe how to use Process Designer (a BPM application) to design your business processes. In addition, we provide step-by-step instructions on how to implement a use case business process scenario (an auto claim approval process).

BPM supports integration with external systems and services through Component Integrator, Web services, and Rules Connectivity Framework. In this book, we discuss these integration options and provide instructions on how to implement the use case by demonstrating some of the options.

At the end of the book, we discuss, from a high level, the planning and designing of content centric BPM applications.

This book is useful for system architects, process analysts, and designers who require an understanding of IBM FileNet Business Process Manager. It also serves as a practical guide for those who want detailed instructions in order to implement a BPM system.

The team that wrote this book

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

Wei-Dong Zhu (Jackie) is an Enterprise Content Management (ECM) Project Leader with the ITSO in San Jose, California. She has more than 10 years of software development and management experience in accounting, image workflow processing, and digital media distribution. Jackie holds a Master of Science degree in Computer Science from the University of Southern California. She joined IBM in 1996 and has managed and led the production of numerous Enterprise Content Management Redbooks in the area of IBM Content Manager, OnDemand, CommonStore, Document Manager, Records Manager, WebSphere® Commerce Module, IBM Content Analyzer (previously known as OmniFind™ Analytics Edition), IBM FileNet Content Manager, and more.
Eric Adel is a Senior Technical Instructor with the IBM Enterprise Content Management (ECM) Education group. He has 15 years of software development experience involving artificial intelligence, data mining, content management, and business process management. In the ECM Education group at IBM, he is the current subject matter expert for IBM FileNet P8 Content Manager API Programming and IBM FileNet Business Process Manager API Programming. Eric holds a Master of Science degree in Electrical Engineering and has finished all course work for his Ph.D. in Object Recognition and Artificial Intelligence. His areas of expertise include IBM FileNet P8 content management, business process management, artificial intelligence, and data mining, as well as EAI, Web services, SOA, and XML. He has designed projects for both the J2EE™ and .NET platforms.

William Benjamin is a Solutions Architect and Information Technology Specialist for IBM's Enterprise Content Management software segment. He has over 10 years of experience working in content management ranging from application development to systems architecture, software engineering, quality assurance, customer education, and technical consulting. William is a subject matter expert in the area of Content Integration and has participated or led in the full spectrum of the software development life cycle from idea inception to post implementation support.

Imtiaz A Khan is a Solution Architect and a Certified Consulting Information Technology Specialist for the IBM Enterprise Content Management software segment. He has 26 years of working experience in the imaging and information management industry. He holds a Bachelor of Engineering and Associate degree in Information Technology. Imtiaz joined IBM in 2000. He holds various industry certifications such as Records Management Practitioner from the Association of Information Technology AIIM, as well as MIT accreditation from AIIM and CDIA+ certification from CompTIA.

Mike Marin is a Distinguished Engineer at IBM Software Group and the Product Architect for IBM FileNet Business Process Manager. He holds a Master of Science degree in Computer Science, specializing in Artificial Intelligence and has more than twenty years of experience designing and developing system software. The last eleven years, he has been developing business process management and workflow products, and participating in standard organizations including OMG, OASIS, and WfMC, working on business process management and workflow standards. He has edited and contributed to the definition of several of these standards. Mike is a Fellow of the WfMC and has received the WfMC Excellence Award for his technical contributions to the WfMC standardization efforts.
Mark Yingling is a Certified Information Technology Specialist for IBM's Enterprise Content Management software segment. He has 20 years of information management and software development experience. He holds a Bachelor of Science degree in Computer Science from Colorado State University. Mark joined IBM in 2001. He is certified by the Association for Information and Imaging Management as an Enterprise Content Management Practitioner.

We would like to extend special thanks to the following people who have made this project possible:

Richard Palfini
Steve Mason
Manoj Puthenveeti
Darik Siegfried
IBM Software Group, Costa Mesa, California

We also thank the following people for their contributions to this project:

Charles Burnett
Jingli Chen
Khoi Dang
Quynh Dang
Victor Falco
Kevin Fally
Evangeline Fink
Steven Kelly
Patrice Lewis
Jane Luo
Mark Muller
Ken Rong
Michael Samiley
Bernadine Sisneros
Chuck Snow
Christopher Timbreza
IBM Software Group, Costa Mesa and Sacramento, California

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.
Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- Use the online Contact us review Redbooks form found at:
  ibm.com/redbooks
- Send your comments in an e-mail to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400
Overview

In this chapter we provide a high level overview of the IBM FileNet Business Process Manager (BPM) and an introduction to the IBM FileNet P8 family of products.

We cover the following topics:

- Business process management
- IBM FileNet Business Process Manager overview
- Process design with BPM
- BPM application development options
- BPM integration to external systems
- BPM process monitoring and analysis
- BPM applications and tools summary
- The IBM FileNet P8 family of products
1.1 Business process management

A business process is a set of activities performed by people and machines to achieve a business goal. Business processes have been used as long as businesses have been in existence because every aspect of a business involves processes. Organizations achieve their objectives by implementing and controlling a diverse set of business processes. Most of these processes are not automated, or are only partially automated.

Business process management involves techniques and tools to manage and automate business processes that are performed by people, applications, and external sources. Automating and streamlining business processes helps organizations to get the work done effectively and efficiently. Managing, transforming, and optimizing business processes are the primary focus of business process management.

An ideal business process management solution should provide the following basic functions:

- Process automation
- Workflow
- Integration with external systems and applications
- Process optimization

Process automation
Processes are performed by people and applications. Automated processes save time when compared with manual processes. The goal of process automation is to provide technology to coordinate how people and applications work together to perform processes. The technology should provide enough functionality and be robust enough to be trusted with business critical processes.

Process automation can be achieved by automating a set of activities in a process using custom applications or business process management products.

Workflow
Workflow is a structured way of performing tasks by an individual or a team. Examples of workflow include auto insurance claims that require the efforts of multiple people to process. The workflow process involves a policy holder filing a claim, a field agent processing the claim, an adjustor adjusting the claim amount, and finally the claimant either receiving the claim amount or the claim being rejected.
A business process management solution should provide the workflow capability. In addition, it must also provide the management and simulation of processes, content integration, and the ability to interface with external systems.

**Integration to external systems and applications**
Integration to external systems and applications enables processes to interact with automated applications and systems that are already in place. The integration leverages the investment in existing applications and systems, and enables quick and easy deployment of business process management applications.

**Process optimization**
Process optimization refers to measuring, monitoring, simulating, and analyzing business processes, and ultimately improving the processes. By providing the continuous process optimization capability in a business process management solution, organizations can increase their responsiveness, identify issues before they become problems, and maximize the performance of their business operations.

### 1.1.1 Transactional processes and content centric processes

Business process management applications generally fall under two categories:
- Transactional processes
- Content centric processes

Transactional processes heavily rely on application integration and system-to-system processing. These are processes that interconnect with the enterprise applications. The processes are short lived and might not require human interaction.

Content centric processes are highly collaborative and require users or systems to make decisions based on content during the processes.

*Content* is any unstructured information ranging from simple e-mails, pictures, documents, and instant messages to complex forms, contracts, and reports. Content can include any file format such as scanned images and electronic documents, rich media, XML, and even business processes.

Content centric processes involve people and content and are usually longer lived than transactional processes. Any addition or change to content, such as new document creation, can trigger the associated process to act on the content, thus the processes are considered content centric.
Most applications require both content and transactional processes to run together to achieve a business goal. As organizations start automating processes, they require both types of processes.

1.2 IBM FileNet Business Process Manager overview

IBM FileNet Business Process Manager (BPM) enables organizations to create, modify, manage, simulate, and analyze business processes. One of the key advantages of IBM FileNet Business Process Manager is its ability to work with active content.

*Active content* refers to the ability of content to trigger or affect business processes. For example, content such as a loan application or a claim application, received via fax, e-mail, or Web, can trigger a process in IBM FileNet Business Process Manager to process the loan or claim automatically. Active content includes the delivery of information in a way that is more unified, accurate, and contextual to the business processes. It enables a wide variety of process management scenarios.

IBM FileNet Business Process Manager can be used in many areas, including banking, insurance, manufacturing, health industry, and government. Examples in the banking and insurance industries include applications for processing loans, claims, credit card approvals and policy underwriting.

IBM FileNet Business Process Manager addresses the primary focus of business process management (that is, managing, transforming, and optimizing business processes). It provides the following functions:

- Process automation
- Process modeling and designing
- Process integration
- Process monitoring and optimization

**Process automation**

IBM FileNet Business Process Manager (BPM) provides a *Process Engine* and a set of associated components and applications that enable easy implementation of robust business process management solutions designed for use by business users. BPM also provides a set of APIs to enable custom programming for automating processes and a flexible environment for process improvements.

**Process modeling and designing**

IBM FileNet Business Process Manager makes a distinction between process modeling and process designing. Process modeling is done for any business
process in the organization, including manual processes that are not expected to be automated. The goal of process modeling is to document and gain an understanding of the process. Process designing, on the other hand, is used to create executable processes.

Process models can be created by tools such as WebSphere Business Modeler, or Microsoft® Visio®. WebSphere Business Modeler is a tool specifically designed for process modeling and to simulate those processes. Microsoft Visio is a drawing tool that can be used to document business processes. In either case, the modeled processes can be used in IBM FileNet Business Process Manager.

IBM FileNet Business Process Manager offers Process Designer, a tool that creates executable processes. It can import Microsoft Visio diagrams and read XML process definition language (XPDL) files that are generated by WebSphere Business Modeler. In addition, Process Designer can read XPDL files generated by other tools. Process Designer can use these models as a starting point to create executable processes. IBM FileNet Business Process Manager also offers Process Simulator, a tool that is used to simulate these processes to identify bottlenecks and to play with “what if” scenarios.

We discuss process design in more detail later in this book.

**Process integration**

IBM FileNet Business Process Manager provides process integration capabilities with external applications and systems through Component Integrator, rules engine framework, and Web services. These integration options accelerate the speed of business process management solution development and reduce the overall costs associated with the development and maintenance of processes.

We discuss process integration in more detail later in this book.

**Process monitoring and optimization**

Business processes have to be managed and monitored so they deliver the desired business goal. The monitoring of business processes can be done on two levels; either at the individual process instance or across all processes. An example of an individual process is the mortgage process for account 123. Monitoring the instance for account 123 will show the current state of that process instance.

Individual process instances can be monitored in several ways. IBM FileNet Business Process Manager includes a tracker tool that displays the status of a particular process in a graphical environment similar to the Process Designer. There is also a milestone feature that can be used to define process milestones during the process modeling. This can be used to monitor the status of the
process. In the mortgage case, some milestones might include application received, in process, approved, and rejected. In this scenario, a business analyst can be assigned as the tracker for a particular mortgage process and the customer can be provided with access to the milestone view of the process.

For monitoring and analysis of all the processes, IBM FileNet Business Process Manager offers Process Analyzer and Business Activity Monitor (BAM). Process Analyzer enables users to evaluate process workload, discover processing trends, and identify bottleneck areas. Business Activity Monitor lets managers monitor various aspects of their business operations, set thresholds, and automate actions to react to thresholds being exceeded.

Information collected by process monitoring can be used for optimization. For newly designed processes, simulation is a technique that can be used to play with “what if” scenarios and detect bottlenecks before placing the process into production.

IBM FileNet Business Process Manager offers Process Simulator, which can be used to analyze processes and evaluate how they behave in different scenarios. Process Simulator can be populated with data from Process Analyzer to work with processes that are already in production.

1.3 Process design with BPM

Process design starts by identifying the activities and the order in which those activities must be performed to accomplish the business process. The outcome of this is a graph that documents the business process and can be used to automate it. IBM FileNet Business Process Manager offers the Process Designer tool, which can be used to design processes and to import processes from WebSphere Business Modeler and Microsoft Visio.

Process design and process improvement are iterative activities because business processes must evolve and adapt to changing business requirements. IBM FileNet Business Process Manager supports the iterative activities by allowing for multiple versions of a process and providing easy management of process evolution.

1.3.1 Process definition

For process models to become executable in the Process Engine, it is necessary to add execution details. An executable process model is called a process definition in BPM and it is stored in an XML Process Definition Language
A process definition describes the activities and resources required to accomplish a business process. It consists of a directed graph, with a series of process activities or steps (nodes) connected together by a series of routes (arcs), which define the sequence in which the steps are executed. Steps and routes are organized into reusable maps. Maps, steps, and routes can be annotated with comments. See Figure 1-1.

![Process Designer showing a map with steps, routes and a step annotation](image)

There can be multiple process definitions per business process to support multiple versions. Process definitions can also inherit from other process definitions. Inheritance allows organizations to specialize processes to create consistency across a set of related processes. The process definitions are stored in the IBM FileNet P8 Content Engine repository.

### 1.3.2 Steps

Steps represent specific business or system activities. Activities can be performed by an individual user (the participant), by a group of users, or by an automated application.
There are several types of steps:

- **Launch step**: The first step in a process. Every process has this step.
- **General step**: Represents a general activity to be processed by a participant (or a group of users), or an automated process. It can be categorized as follows:
  - **Participant step**: A step that has an associated participant or a group of users, all of whom must process the work item to complete the step. The identity of these users can be defined at runtime through the use of groups.
  - **Work queue step**: A step that is assigned to a *work queue* (see definition below) instead of a specific participant.
  - **Unassigned step**: A step that has no effect in the process and can be used for routing or documentation purposes.
- **System step**: Represents one or more functions to be performed by the system. For example, a system step might include assigning data field values, creating a new process instance, or suspending a process for a specified period of time.
- **Submap step**: Calls another map in the current process definition.
- **Component step**: Performs operations in an external application or system. It is accomplished by IBM FileNet Business Process Manager’s Component Integrator.
- **Web services step**: Invokes or implements Web services. IBM FileNet Business Process Manager uses Web services to integrate to external applications and services.

**Work queue**

A work queue holds work items that can be completed by a user from some user groups or by an automated process. Assigning work to a work queue rather than to an individual participant provides flexibility as to who can perform the specific activity at a step.

### 1.3.3 Routes

*Routes* define the order of execution for a series of steps based on specific rules (using workflow data fields) and user responses (Approve and Deny as examples). You can specify to always take a route or to only take the route when a condition is met.
Routes can be used to define optional branches in a process map. For example, in Figure 1-1 on page 7, the Evaluate step has two optional outgoing routes, one labeled Approved and the other Reject. If a user approves the application, the Approved route is taken. If the user rejects the application, then the Reject route is taken.

Routes can also be used to define parallel branches where the steps are executed in parallel.

1.3.4 Maps

A map represents the sequences of steps and routes required to complete a process. You can define maps using Process Designer or import them from other process design tools, such as Microsoft Visio.

A complex process can be broken down into simpler and reusable components with each component being represented by a map. A process, therefore, can use a collection of maps. The ability to break down processes into smaller components makes complex processes easier to build and understand. It also results in a significant reduction in the total cost of development. The reuse of the maps ensures the consistency of processing and reduces time and costs associated with the deployment of new processes.

A process definition that has been deployed can be executed by another process. Processes that are used this way are called subprocesses. Subprocesses are different than maps in that subprocesses are complete processes, while maps are process segments.

1.3.5 Deadlines and timers

Users can create deadlines and timers to ensure that work is processed in a timely manner. A deadline provides a time-based scheduling constraint, which requires that a step or process be completed within a certain amount of time. The deadline can be relative to the time that the step was routed to the participant, or to the time that the process was launched.

A participant with a deadline can receive a reminder of the pending deadline through e-mail. When the deadline is passed, a visual reminder is displayed in the participant’s inbox, and an e-mail can be sent to a group of users.

A timer specifies a period of time before it is triggered. Multiple timers can be created, terminated, and modified during a process. A timer specifies a sub-map that is executed when the timer expires.
1.3.6 Content events

Processes can be automatically launched when objects in a content repository are created or changed. In general, any content event can be used to launch or to trigger a process. For example, if a new customer (object) is created, a process that manages new customers can be automatically launched.

1.4 BPM application development options

There are several ways in which P8 Web applications can be developed using one or a combination of the following development options:

- IBM FileNet Workplace™ or Workplace XT¹ are powerful J2EE applications that can be customized to incorporate business logic.
- IBM FileNet Business Process Framework (BPF) allows customers to develop and deploy quickly and efficiently a fully customized Web application with interactive user interface and no additional coding.
- Electronic forms (eForms or Lotus® Forms) provide build-in business logic, integration with back-end systems with data validation, and an easily customizable user interface that enables quick development and deployment of form-centric business processes.
- Portal applications can be provided through portlets.
- Web Application Integration Toolkit (WAT) provides an extensible framework of reusable modules that facilitates the development of HTML based applications.
- Process Engine API provides access to all BPM functionality.

1.4.1 IBM FileNet Workplace (or Workplace XT)

IBM FileNet Workplace (or Workplace XT) is a Web application for users, process designers, and system administrators that provides enterprise content management and business process management functions. Workplace comes with a rich set of content centric functions including the ability to browse and search for documents and process in-boxes, and provides access to IBM FileNet Business Process Manager tools such as Process Designer and Tracker.

¹ Workplace XT is the latest release of the Workplace application. Although this book deals exclusively with Workplace, many of the features and functions that are discussed for Workplace are also applicable for Workplace XT. For simplification, we mention just Workplace in many parts of the book.
Workplace can be used as-is. You can also customize it for your business requirements, or use it as a sample for custom application.

1.4.2 Business Process Framework (BPF)

Business Process Framework (BPF) provides a highly configurable framework for rapidly developing IBM FileNet Business Process Manager applications. It presents IBM FileNet Business Process Manager functions based on a case management model. A case involves documents and processes that are used in conjunction for a business application.

BPF supports a broad set of functions, including data entry, layout with built-in validations, and data types. An IBM FileNet Business Process Manager Web application can be created with simple configuration rather than coding. The framework provides user interface components that can be configured and customized easily, thus reducing development time and costs.

As a rapid application development tool, BPF is generally used in implementing IBM FileNet Business Process Manager systems where time is a critical factor. BPF is also a good fit in situations where business requirements are unclear and are changing frequently, so an agile solution development is necessary. BPF is very flexible in adapting to changes in the business requirements.

1.4.3 Electronic forms (eForms or Lotus Forms)

Forms increase productivity by minimizing the amount of time spent in creating user interfaces and ensuring a rapid user acceptance through the visually rich eForms or Lotus Forms user interface.

The user interface for a particular step in a process can be created using the eForms or Lotus Form Designer. The resulting forms can be automated and tightly integrated with the Process Engine without custom programming.

The form filled out by a user in a process step can be saved and stored in the content repository as a document, which can then become a record of the user action. In addition, data in the form can be used to populate the document metadata or update an external database.

Forms provide features such as automatic calculations, data validations, and error messages to assist users, and to help prevent data entry errors.
1.4.4 Portal user interface through IBM FileNet P8 portlet

IBM FileNet P8 provides a set of predefined portlets that are compatible with industry standard Web portals. A portlet is a user interface component that encapsulates a small set of functions.

IBM distributes the source code for the predefined portlets, so their capabilities can be modified and extended. In addition, users can create their own portlets using the Content Engine and Process Engine Java™ APIs.

1.4.5 Web Application Toolkit (WAT)

Web Application Toolkit (WAT) provides an extensible framework and reusable modules for building Web applications. The WAT reusable user interface component facilitates the development of a robust Web-based application with a user interface that requires little or no DHTML/JavaScript™.

WAT includes a rich set of APIs that can be used to create custom Web applications or to integrate IBM FileNet P8 components with existing applications. IBM FileNet Workplace is an example of a Web application that has been developed using the toolkit.

1.4.6 Process Engine API

IBM FileNet Business Process Manager provides a full set of APIs, called Process Engine APIs, which enable you to create applications that can take advantage of all the IBM FileNet Business Process Manager functions. The APIs provide configuration, management, and execution functionality. All the IBM FileNet Business Process Manager tools and applications are written using the same APIs exposed to you.

1.5 BPM integration to external systems

Access to external systems, applications, or databases can be established in various ways, including but not limited to IBM FileNet P8 Component Integrator, Web services, and business rules engines.

1.5.1 Component Integrator

Component Integrator makes it possible to interact with an external entity, a component such as a Java object, or Java Messaging System (JMS) from a
process step. It includes adaptors to communicate events from the Process Engine to external entities such as Java objects.

**Note:** IBM FileNet Business Process Manager consists of Process Engine, Content Engine, and Application Engine. For information about these engines, refer to Chapter 3, “System architecture” on page 35.

### 1.5.2 Web services

Web services are self-contained, self-described, modular services that can be published, located, and invoked across the Web. In a typical Web services scenario, a business application sends a request to a service at a given URL over HTTP. The service receives the request, processes it, and returns a response. With IBM FileNet Business Process Manager, a business process can both send and receive Web services calls. Steps in a process can invoke Web services.

Web services calls can also start a business process in IBM FileNet Business Process Manager. A business process can be exposed as a Web service.

### 1.5.3 Business rules engines

Business rules engines allow customers to extract and manage business rules outside the applications and processes. The rules framework allows business rules engine vendors to integrate their engines into IBM FileNet Business Process Manager.

In a process, business rules can be used in two distinct ways. The first is to control the process itself. For example, steps can be skipped or re-executed. The second is to implement application logic at a particular step. For example, a business rule can be used to calculate the price of a discounted item.

The IBM FileNet P8 Platform can be integrated with industry standard rules engines enabling process designers and business analysts to associate rules with individual steps in a process.

### 1.6 BPM process monitoring and analysis

IBM FileNet Business Process Manager provides three ways to analyze business processes: Process Simulator, Process Analyzer, and Business Activity Monitor (BAM). The Process Simulator is used to analyze the process during the
development phase. The Process Analyzer is used for dynamic, chart-based analysis of process activity at runtime. The IBM FileNet Business Activity Monitor is used to monitor various aspects of in-progress business operations including not only the executing processes but associated applications.

As shown in Figure 1-2, the life cycle of process improvement starts by analyzing the current process that can be manual or automated, and identifying areas for improvement. Simulation technology can be used to validate the process. Then the improved process can be deployed into production, where it can be monitored using Process Analyzer or BAM.

1.6.1 Process Simulator

Process Simulator is an application (tool), bundled with IBM FileNet Business Process Manager, that is used to validate a process before placing it into production. There are two scenarios where users would use simulation. The first case consists of validating a brand new process, where the user provides the information for the simulation based on experience and knowledge of the process domain. The second case consists of validating enhancements to an existing process, where historical data from the BPM repository is used to feed the simulation. In either case, the user would simulate what-if scenarios and then analyze the results of the simulation to validate the process. That way, an analyst can test different scenarios to improve the business process before deploying it in a production environment.
Chapter 1. Overview

Figure 1-3 shows a process simulation user interface.

Figure 1-3    Process simulation

1.6.2 Process Analyzer

Process Analyzer is an application (tool), bundled with IBM FileNet Business Process Manager, that is used to discover business processing trends, to establish a baseline for productivity measurement, and to identify bottlenecks. Process Analyzer monitors and analyzes events generated by the Process Engine. The Microsoft Analysis Services, an On-Line Analytical Processing (OLAP) solution, is used to store the aggregated results of the analysis. Process Analyzer uses Microsoft Excel® as the default reporting tool for viewing and analyzing the data stored inside the OLAP database. It uses Excel as a low-cost common application that has a rich set of chart and reporting functions.

You can create your own custom reports, extending the reporting functions by including business data that is tracked in the Process Engine event logs. In addition to Microsoft Excel, if you require sophisticated reporting capabilities, you can use IBM Cognos® Analysis Studio or IBM Cognos Reports.
1.6.3 IBM FileNet Business Activity Monitor

IBM FileNet Business Activity Monitor (BAM) is based on IBM Cognos technology, and can be used to monitor various aspects of the in-progress business operations in almost real-time. You can set thresholds and automate actions to react to thresholds being exceeded. The business operations can be steps that are managed by the Process Engine or activities that are managed by other enterprise applications. You can configure BAM to collect and correlate data from various data sources and generate a holistic analytical view of the business operations.

Key Performance Indicators (KPIs) are financial or non-financial performance parameters that organizations set based upon their corporate guidance. You can set thresholds that reflect KPIs and configure BAM to send alerts and perform actions if thresholds are violated.

Figure 1-5 shows the BAM Dashboard which illustrates various business activities such as case arrivals, arrivals by hour, cases by product, along with respective service level agreement (SLA) performance.
1.7 BPM applications and tools summary

Here is a summary of the IBM Business Process Manager applications and tools:

- Process Designer
- Process Simulator
- Process Analyzer
- Tracker (also known as Process Tracker)
- Configuration Console (also known as Process Configuration Console)
- Task Manager
- Workplace or Workplace XT
- Business Process Framework
- Electronic forms (eForms or Lotus Forms)
- Business Activity Monitor

The last three items on the list do not come with IBM FileNet Business Manager. They are separate components that work with and support IBM FileNet Business Manager solutions (as we have covered briefly in previous sections of this book).
The main tool to do process design in IBM FileNet Business Manager is Process Designer. We discuss how to use this tool in more detail in Chapter 5, “Using Process Designer” on page 67. Although other supporting applications and tools are also important in improving, monitoring, and managing your business process, it is beyond the scope of this book to cover them.

1.8 The IBM FileNet P8 family of products

In addition to IBM FileNet Business Process Manager, the IBM FileNet P8 family of products includes back-end services, development tools, and applications that address enterprise content and process management requirements. The IBM FileNet P8 Platform enables users to automate business processes, access and manage all forms of content, and automate records management to help meet compliance requirements. It is a reliable, scalable, and highly available enterprise platform that enables users to capture, store, manage, secure, and process information to increase overall operational efficiency.

The IBM FileNet P8 Platform uses an open, standards-based J2EE architecture, and is designed to fit in a variety of environments including Microsoft.NET. It supports a large selection of databases, applications, operating systems, portals, security frameworks, storage, systems management tools, and Web server environments.

The IBM FileNet P8 family of products are divided into three major categories:

- Content
- Process
- Compliance

There is also a set of tools that the IBM FileNet P8 Platform provides for system management.

1.8.1 IBM FileNet content products

The IBM FileNet P8 family of products provides comprehensive enterprise content management functions and offers a number of products in the content management category.
IBM FileNet Content Manager

IBM FileNet Content Manager manages highly customized and complex content types while maintaining secure control over metadata and compliance activities. At the core of IBM FileNet Content Manager are the repository services for capturing, managing, and storing your business-related digital assets. Multiple repositories, called object stores, can be created and managed within a single system to serve the business requirements. Object stores can be configured to store content in a database, a file system, a fixed content device (such as an IBM FileNet Image Services repository, Network Appliance™ SnapLock®, or Centera), or a combination of these options.

An object store is capable of storing a variety of business-related data — for example, an auto insurance claim, a customer mortgage loan account, or information about business partners. It can also store any type of structured or unstructured content such as XML documents, Microsoft Office documents, Web pages, photos, voice data, images, process definitions, templates, and more.

Content Federation Services (CFS)

CFS provides the capabilities to integrate with and manage content in both IBM FileNet and non-IBM FileNet content repositories.

Content Federation Services for Image Services (CFS-IS)

Content Federation Services for Image Services (CFS-IS) natively integrates and federates content from Content Engine and Image Services repositories. CFS-IS enables Content Engine to use Image Services as another content storage device. Users of IBM FileNet P8 applications have full access to content stored in existing Image Services repositories.

Anything that is created in Workplace or created programmatically via the Content Engine APIs can be stored in the Image Services permanent storage infrastructure. Existing Image Services content is preserved and usable by Image Services applications, and reusable by the IBM FileNet P8 family of products, such as IBM FileNet Workplace and IBM FileNet Records Manager, without duplication and without change to existing applications. The location of document content is transparent to all applications.

1.8.2 IBM FileNet process products

The IBM FileNet P8 family of products provides enterprise process management. The products offered in this category include IBM FileNet Business Process Manager, IBM FileNet Business Process Framework, IBM FileNet eForms, and IBM FileNet Business Activity Monitor. For a brief description of these products, see 1.4, “BPM application development options” on page 10 and 1.6, “BPM process monitoring and analysis” on page 13.
1.8.3 IBM FileNet compliance products

IBM offers a cohesive compliance framework that combines e-mail and file system capture with records policies and BPM processes. Using the IBM FileNet P8 family of products, compliance can be enforced automatically without user intervention.

There are three core products under the compliance framework: IBM FileNet Records Manager, IBM FileNet Records Crawler, and IBM FileNet Email Manager.

**IBM FileNet Records Manager**

The IBM FileNet P8 Platform and IBM FileNet Records Manager provide the software components together in a single system architecture to assist organizations in achieving records management compliance, with no user involvement. The completely automated declaration and classification of records is called *ZeroClick* in the IBM FileNet compliance framework. IBM FileNet Records Manager helps solve regulatory compliance and records management issues by helping organizations easily capture, organize, monitor, retrieve, and retain records for improved decision making and adherence to regulatory requirements.

Records can be automatically captured, declared, and classified, using any one of the following methods:

- **Document centric**: The declaration and classification of records are based on documents. When a file is saved to a particular folder in the IBM FileNet P8 content repository, the system responds to certain criteria to declare and classify the document as a record. The process can be set up to be completely transparent to users, eliminating the necessity for additional steps or manual decisions associated with records declaration and classification.

- **Event based**: The declaration and classification of records are based on events. Following certain events or transactions associated with Web-based or line-of-business applications, records can be automatically captured and classified, based on pre-defined criteria or rules, such as the type of transactions or metadata associated with the transactions.

- **Process-centric**: The declaration and classification of records are based on business processes. At pre-determined points in a business process, records can automatically be identified, declared, and precisely classified, based on pre-defined criteria or data from the business process.

IBM FileNet Records Manager reduces the risk of litigation and provides business continuity by enforcing corporate compliance procedures, storing only records that are required and only for as long as they are required, and ensuring that expired records are destroyed in a legally acceptable manner.
IBM FileNet Records Crawler
IBM FileNet Records Crawler is a capture tool (or application) that scans through file systems and, based on specified rules, captures and stores electronic files into Content Engine where they can be declared and classified as records.

IBM FileNet Email Manager
IBM FileNet Email Manager is similar to IBM FileNet Records Crawler but processes e-mails on Exchange, Domino® and Groupwise e-mail servers. It helps organizations easily capture, organize, monitor, retrieve, retain and share e-mail content for improved decision making and adherence to regulatory requirements.

1.8.4 IBM FileNet system management tools

The IBM FileNet P8 Platform offers tools that provide monitoring of system components and applications such as the Process Engine, database, IBM FileNet P8 Image Services. The system management tools include IBM FileNet P8 System Monitor and IBM FileNet System Manager.

IBM FileNet System Monitor (FSM)
The IBM FileNet P8 Platform supports enterprise wide and mission critical applications in which the availability of a system is vital. In order to achieve high availability and maximum uptime, organizations must have hardware and software that supports a high availability and disaster recovery infrastructure. In addition, it is important to have a system to monitor all areas of software operation and report any problems.

IBM FileNet System Monitor automates the monitoring of the entire IBM FileNet P8 environment, including software components, application servers, databases, log files, network communication and devices, and the full range of IBM FileNet storage libraries.

IBM FileNet System Manager
IBM FileNet System Manager is a tool for collecting and distributing performance data on the entire IBM FileNet P8 family of products, independent of the operating system. The performance data gathered is available through a dashboard.
Solution example

In this chapter we describe a scenario where IBM FileNet Business Process Manager is used to optimize and streamline an existing business process that is mostly manual, slow, and error-prone.

We cover the following topics:

- Scenario background
- Claim approval process
- Summary of benefits
2.1 Scenario background

Consider a fictitious company, XYZ Corporation. It is a large financial services organization that offers a wide range of financial products and services through a large network of field offices. They are headquartered in the United States with satellite offices worldwide.

XYZ Corporation is looking to improve business processing and streamline inefficiencies in their claim approval process. Their current process is partially manual, time intensive, and prone to errors.

The current process is dependent on the documentation that comes from the field offices. If there is a mistake, such as missing signatures, or if a document takes too long to get into the office, it impacts the entire claim approval process. Due to the mostly manual process, it takes a long time to get claim settlements out to the policy holders. In addition, it is difficult to find and group all of the documents associated with a policy holder and their open or closed claims, and forms often get lost in the mail or in the mail room. With a long and error-prone claim approval process, XYZ Corporation faces unsatisfied customers, and the risk of losing customers and market share.

The process is content intensive and requires people to make decisions based on that content; therefore, IBM FileNet Business Process Manager is the perfect solution to streamline this business process.

2.2 Claim approval process

Let us examine how the claim approval process works in the XYZ Corporation and see how IBM FileNet Business Process Manager can improve the overall business process.

A policy holder has an auto insurance policy with XYZ Corporation. The policy holder has been involved in an automobile accident and submits a claim with the XYZ Corporation. Once the claim is submitted, it will either be approved or denied. If the claim is approved, the policy holder will get a check to cover repairs to their vehicle and can secure a rental car while their car is being repaired. If the claim is denied, the policy holder will be notified and will not get paid.
Figure 2-1 shows the business context diagram for the claim approval process. It illustrates the sequence of steps (numbered) throughout the entire process, the people involved in the process, and how content (a claim package) flows through the process to be used or processed.
When a customer submits a claim to the XYZ Corporation, several questions arise within the business process and business practice, including these:

- Will the claim be approved or denied?
- If the claim is denied, what was the reason?
- How does supporting documentation get to the right person at the right time?
- What are the factors that help the XYZ Corporation make decisions more efficiently and accurately on approving or denying claims?
- How can the turn-around time for the business process be more efficient?

Using effective business process management practices, with IBM FileNet Business Process Manager, helps to automate and streamline the claims approval process. Let us look into the individual stages of the process and see how BPM improves this overall business process.

Based on Figure 2-1, the claim process can be broken down into the following logical stages:

- Initiating a claim (steps 1 and 2)
- Back office mailroom clerk task (step 3)
- Claim processing (steps 4 through 6)
- Financial controls (part of steps 5 through 6 and 7)

### 2.2.1 Initiating a claim (steps 1 and 2)

An insurance claim is initiated at XYZ Corporation when a call is placed to a Customer Sales Representative (CSR) from a customer. This is step 1 in Figure 2-1. The customer (or the claimant) should be a policy holder whose policy is in force. The CSR (also called agent for convenience) who takes the call from the claimant has the responsibility to verify information about the claimant as well as information about where and when the incident occurred. The agent fills out all of the necessary forms to track this information and then faxes the documents to the XYZ Corporation back office for processing. This is step 2 in Figure 2-1 on page 25.

The current process to initiate a claim is manual and error-prone. It relies on manual forms and paperwork to be filled out by an agent. The filed form and paperwork for the claim are essential for starting the process, which is driven by that content. In the current manual process, the content is passive: The agent has to file the form and fax the information over to manually start the claim process. In the automated solution, the content should take a more active role in driving the business process.
Improvements using BPM technology

Business processes in IBM FileNet Business Process Manager are capable of interacting with content in various stages. IBM FileNet Business Process Manager allows business processes to monitor and subscribe to content in a repository. If a new piece of content is stored in the repository, or an old piece of content is changed, a business process can be launched automatically. It also allows content to be attached to the business processes. This makes the content aware of the business processes. Using IBM FileNet Business Process Manager, we no longer treat content and process as separate items; we integrate them to create more effective, streamlined business process management.

The agent takes the call from the customer and creates documentation about the incident, such as where the auto loss occurred and when. The agent also faxes the documentation about the loss to the back office. The faxed information creates a document on the receiving end with its own life cycle and retention rules. This is the content that is crucial to the approval process. The information that is documented by the agent is what starts the claim approval process. Because this filled-out document starts the claim approval process, it is clear that when a new document of this type is created, a new claim process could be started automatically. The combination of creating the initial claim document, populating it with information, and faxing the document to the back office are the key pieces that start the claim approval process.

We have identified the components that start the process. Now we can streamline the process and optimize the workflow by unlocking the value of the content. Leveraging components such as electronic forms (eForms), we can automate the paperwork that the agent is required to fill out to initiate the claim. Leveraging database lookups within the electronic forms, we can pull up the customer's information such as the address, make and model of the vehicle, and the VIN number.

The electronic form can take items such as customer number, policy number, or customer name, and retrieve the additional information required to start the claim approval process. The initial claim document can be combined with additional supporting documents about the claim and stored in the content repository together. If there is ever a need to look up and review any of the customer’s documents, we can find them all filed in the same system and using a single search interface. IBM FileNet Business Process Manager leverages the stages and life cycle of the content and makes it an active part (active content) of the business process.
**Features and capabilities of BPM utilized**

The features and capabilities of IBM FileNet Business Process Manager utilized at this stage are:

- Electronic form filling (eForms)
- Database lookup
- Active content (electronically filled claims), which automatically triggers a new claim approval process

**2.2.2 Back office mailroom clerk task (step 3)**

The XYZ Corporation back office works on the claim after it has been created by an agent. After an agent fills out the necessary paperwork about the customer's loss, the information is sent to the back office. The back office's mailroom clerk receives the claim via fax, timestamps the claim package, searches for a policy document, and gets the name of the field agent based on the location of loss. Finally, the back office mailroom clerk sends the claim package to the field agent via US mail. This is step 3 in Figure 2-1 on page 25.

**Improvements using BPM technology**

Content is an important driver in this step in the process. The information about the customer and the loss must be documented and classified. The claim package must be forwarded to a field agent for review and process.

IBM FileNet Business Process Manager has a *routing* capability that facilitates the routing of content to other users, processes, or systems. In addition to automatically routing the claim package to a field agent, we can also leverage the conditional routing capability to assign the right field agent to handle the claim, based on data such as the policy holder and the location of the loss. IBM FileNet Business Process Manager can remove time consuming tasks such as interoffice mail routing, manual lookup, and assignment of field agents. We can leverage *database lookups* and *Web services* to find out how to get the right information to the right person at the right time.

**Features and capabilities of BPM utilized**

The features and capabilities of IBM FileNet Business Process Manager utilized at this stage are:

- Electronic routing of work packets (claim packages) to the right people
- Database lookup
- Automation of tasks (such as field agent assignment based on business rules)
- Web services to get the right information to the right person at the right time
2.2.3 Claim processing (steps 4 through 6)

A field agent receives the claim package. The field agent is responsible for reviewing the claim package and checking for completeness and errors in the claim. As part of the claim process, the field agent must gather supporting documents about the customer's loss. This supporting information can consist of eye-witness statements, pictures from the loss location, and police statements. The field agent is responsible for getting the supporting documents and content to the adjustor. After the supporting documents are collected, they have to be filed with the other documentation that has already been created for the claim. This is *step 4* in Figure 2-1 on page 25.

Delays in getting the supporting documents to the adjustor can slow or halt the claim approval process. After receiving the claim package, the adjustor reviews the entire claim package and decides to either approve or reject the claim. Alternatively, the adjustor escalates the claim to a supervisor. The approved claim goes to the accounting department for payment and record keeping. This is *step 5* in Figure 2-1 on page 25.

The adjustor might rely on several external systems to get all of the information required to complete the customer's claim. Working with the field agent to process the claim, the adjustor uses the supporting documents to determine the validity of the loss or the amount of monetary compensation that the policy holder is entitled to receive.

If the claim escalates to the supervisor, the supervisor either approves it or rejects it. The approved claim goes to the accounting department for payment and record keeping. This is *step 6* in Figure 2-1 on page 25.

The steps in this claim process can be delayed by dependencies on surface and interoffice mail to get the claim packages and loss information to the field agent and the adjustor. Other areas that can slow the process include the utilization of external systems to locate other information required to process the claim. If the field agent does not assign an adjustor when all the required documents are available, the claim process can be further delayed. Automation in assigning an adjustor and getting the right information to the field agent and the adjustor are beneficial in removing latency from these steps.

**Improvements using BPM technology**

IBM FileNet Business Process Manager streamlines this part of the process by providing automated mechanisms for the initial claim packages to get to the field agent and later to the adjustor's desktop. The field agent no longer has to work with faxed documents that might be illegible and require several hours to check for completeness and errors. Validation can be performed at the entry time and at the application level. This way, field agents can be assured that the work is assigned to them with the right data.
IBM FileNet Business Process Manager optimizes the collection of supporting documents by providing the field agent with a mechanism to submit and attach content to the claim folder. When the content is added to the claim folder, IBM FileNet Business Process Manager locates the related documents about the loss and files them in a location that contains all of the content for the claim. This eliminates the necessity for using costly methods such as surface or interoffice mail to get the supporting documents to the field agent and the adjustor.

IBM FileNet Business Process Manager can also automate the classifying and processing of this content, thus ensuring that it is filed in the correct customer folders. This enables fast discovery of content for future claims and can help satisfy record management and compliance requirements.

In addition to the above capabilities, IBM FileNet Business Process Manager can interact with external systems, data sources, and applications to get the information required to process the claim. External systems can provide the customer's information, such as additional policies, previously submitted claims, and pending claims, to the field agent and the adjustor.

**Features and capabilities of BPM utilized**

Features and capabilities of IBM FileNet Business Process Manager utilized at this stage are:

- Electronic routing of work packets to the right people as soon as possible
- Database lookup (for existing policy information and historical data)
- Active content with document attachment
- Foldering (adding documents to a folder or work package)
- Search the content in the system
- Interaction with the external systems and data sources to get the right information for processing

**2.2.4 Financial controls (part of steps 5 through 6 and 7)**

The adjustor examines the claim package and its supporting documentation and determines the monetary amount of the claim. The supporting documentation, repair estimates, notes, and annotations that have been added throughout the process are all relevant to making a determination of the amount of the loss. In addition, the witness statements and police reports determine whether the customer is liable for the loss.

The supervisor informs the adjustor if the claim is rejected. The adjustor then informs the field agent of the decision. A letter outlining the reasons for the rejection is generated and it is sent to the customer. The field agent notifies the customer (either by phone or email) of the decision.
If a claim is approved, the adjustor or the supervisor sends the claim to the accounting department. The accounting department generates a payment and sends it out to the customer. A letter is generated and sent, notifying the customer about the approved status and stating when the payment will be sent. The accounting department then archives the payment information along with the claim for future audit or retrieval. This is step 7 in Figure 2-1 on page 25.

**Improvements using BPM technology**

IBM FileNet Business Process Manager can integrate electronic forms, content, and business processes to create solutions that are logical and intuitive for users to use. The adjustor uses the content in the claim package to determine whether the claim amount is within the limit to be automatically approved. If the monetary amount of the claim is within a pre-determined range, the adjustor has the ability to automatically approve it. If the claim amount exceeds that limit, the adjustor escalates it to a supervisor. The supervisor has the additional tools and authority to make decisions on whether or not to approve a claim.

Using IBM FileNet Business Process Manager, we can use the *routing* capability to automate the way the work goes from an adjustor to a supervisor. The *routing rule* can be implemented with conditions that make it seamless for work to be routed from the adjustor to the supervisor. After the work is completed (approved by the supervisor if required and by the adjustor), it can resume its progress along its defined path in the workflow, or it can be sent back to the point where it started. IBM FileNet Business Process Manager also has the capability to *escalate* the claim to another person such as a manager, if a certain amount of time has expired (for example, three days).

IBM FileNet Business Process Manager has the ability to *interact with external systems and processes* to automate the creation of letters or the generation of payment for customers. After the letters, invoices, or purchase orders are created, IBM FileNet Business Process Manager can *automatically file* the letter along with the other documents from the claim in one place, making it simple to find all related content items for the customer or the claim.

**Features and capabilities of BPM utilized**

The features and capabilities of IBM FileNet Business Process Manager utilized at this stage are:

- Routing with routing rules to automatically route work packets to the appropriate people
- Wait and resume capability in a business process
- Automation of tasks (such as printing of letters to customers)
- Interaction with external systems (such as interaction with an accounting system for check printing)
2.2.5 Workflow process maps for the claim approval process

With IBM FileNet Business Process Manager, we can create workflow process maps that automate and facilitate the claim approval process.

Figure 2-2 shows the main workflow process map that drives the claim approval process.

![Figure 2-2 Main workflow process map for the claim approval process](image)

The activities of setting up a claim case that involves creation of a case (folder) for the claim, assigning of a field agent, gathering of supporting documents, and assigning of an adjustor are encapsulated in a workflow subprocess, which is implemented as a separate submap. Figure 2-3 illustrates this claim setup submap.

![Figure 2-3 Claim set up workflow process submap for the claim approval process](image)
We cover process design, including the creation of a workflow map and submaps, in more detail in later chapters of the book.

### 2.3 Summary of benefits

In this claim approval process example, we showed that by using IBM FileNet Business Process Manager, we can create the content (claim document) and automatically start a business process (claim approval process). IBM FileNet Business Process Manager allows an organization to build processes that can interact with content to provide higher-value (more accurate and efficient) solutions. It provides the structure required to optimize the claim approval process.

By automating the process and using routing of the work to different people, work (the claim package) can move seamlessly between field agents, adjustors, supervisors, and the accounting department. IBM FileNet Business Process Manager also interacts with external systems, making the generation and distribution of letters to policy holders a simple, automated process. The process is made more efficient and less error prone. Reporting and audit and archival are simplified.

The features and capabilities of IBM FileNet Business Process Manager showcased in this solution example include:

- Electronic form filling (integration with eForms, another IBM FileNet P8 product)
- Active content: A new claim automatically triggering a new business process
- Active content: Document attachment (for the supporting documents) to the existing work packet
- Electronic routing of work packet to the right people as soon as possible
  
  Example:
  - Routing of work packet (claim package) to an adjustor or agent depending on where the step is in the process
  - Routing of work packet based on the claim amount or other routing rules
  - Escalation in routing, if a certain condition is true
- Wait and resume capability for a business process
- Database lookup (for existing policy information and historic data)
- Foldering, filing, and adding documents to a folder or work packet
- Efficient search of the content in the system
- Automation of tasks
  Example:
  - Assigning a field agent to a claim based on the location of the accident
  - Assigning an adjustor to a claim based on business rules
  - Printing a letter of the claim status to the customer

- Web services and interaction with the external systems and data sources to get the right information to the right person at the right time
System architecture

In this chapter we describe the underlying architectures for the IBM FileNet P8 Platform and IBM FileNet Business Process Manager (BPM). We show how BPM technology under the IBM FileNet P8 Platform is leveraged to deliver automated content centric processing to businesses.

We cover the following topics:

- IBM FileNet P8 core engines
- Layered architecture
- Process Engine logical view
3.1 IBM FileNet P8 core engines

There are three primary engines that make up the IBM FileNet P8 Platform. For our purposes, we define an *engine* as a collection of services and components, which perform a set of related functions. Although an engine is composed of many parts, we view it as a single functional unit. Understanding the complexity of how each engine works is not necessary to gain an appreciation for how a business process is defined, developed, and managed in the IBM FileNet P8 environment. However, it is important to know what each engine does.

The three core engines for the IBM FileNet P8 Platform are:

- **Content Engine:** The Content Engine provides software services for managing different types of business-related unstructured content, which we refer to as objects. It manages one or more object stores. An object store is a repository for storing objects in an IBM FileNet P8 environment.

- **Process Engine:** The Process Engine components allow you to create, modify, and manage automated business processes. These processes are performed by applications, enterprise users, or external users, such as partners and clients.

- **Application Engine:** The Application Engine hosts the Workplace Web application, Workplace Java applets, and Application Programming Interfaces (APIs). It is the presentation tier for both process and content.

Figure 3-1 illustrates the three IBM FileNet P8 engines and their relationships.
3.2 Layered architecture

IBM FileNet Business Process Manager is designed around content centric process management and ease of use paradigms. The design goal is to provide a BPM solution that can be easily used by business analysts to take control of the business processes of their organizations. Business analysts are responsible for authoring, managing, and monitoring the processes implemented in the IBM FileNet Business Process Manager solution.

IBM FileNet Business Process Manager is designed as a four-tier architecture as shown in Figure 3-2:

- Client tier
- Presentation tier
- Middleware tier
- Data tier

![Figure 3-2 IBM FileNet Business Process Manager four-tier architecture](image-url)
Most organizations install IBM FileNet Business Process Manager solutions using different servers for each tier, but tiers can be collocated into fewer servers.

In the IBM FileNet P8 Platform, the functionality is exposed using the same tiered architecture with other components in each tier. Those extra components provide valuable features for the BPM functionality, including content management, electronic forms, records management, collaboration functionality, and more. Figure 3-3 shows a high level architecture of the IBM FileNet Business Process Manager components.

Figure 3-3  High level architecture of IBM FileNet Business Process Manager components
3.2.1 Presentation tier

The presentation tier contains most of the IBM FileNet Business Process Manager applications and tools that users need to interact with the system. The APIs, external communication, and customer-written business logic are exposed in this tier. In IBM FileNet P8, this tier is implemented by the Application Engine and it is hosted in a Java Application Server.

**Component Integrator**

The Component Integrator is used to integrate Java or Java Message Service (JMS) components for use in processes. Components are registered in the Process Configuration Console to make them available in Process Designer. Figure 3-4 shows a Java component that has been registered.

![Registered Java component with Component Integrator](image)
**Process Orchestration**

*Process Orchestration* provides the Web services functionality of IBM FileNet Business Process Manager. It provides the ability for processes to invoke Web services (WS), and to receive Web services calls. Most of the process orchestration functionality is implemented in the middleware tier, but the communication part is implemented at the presentation tier.

Figure 3-5 shows the Process Orchestration system architecture.

![Figure 3-5 Process Orchestration system architecture](image)

The Process Orchestration component can be started by the Task Manager and it is composed of a Web services Listener that receives all the incoming messages and a Web services Adaptor that is used to send outgoing messages.
3.2.2 Middleware tier

The middleware tier contains the core of the BPM system as shown in Figure 3-6.

![Middleware tier diagram]

The IBM FileNet BPM middleware tier includes the following components:

- **Content Engine (CE) Kernel**: The core Content Engine.
- **Process auto-launch**: The event-driven component that launches process instances in the Process Engine system. This component can be configured to launch a specific process version in response to specific events. For example, filing a document in a folder can launch a particular process in the Process Engine system.
- **Process Engine (PE) Kernel**: The core engine of the BPM system that includes the execution sub-system.
- **E-mail Notification**: Enables automatic transmission of e-mail to users when specified process-related events occur. E-mail notification can also be used to track processes.
- **WS API**: Provides a Web services API to the Process Engine and to the Content Engine.
- **Rules Framework**: Provides a framework for rules engines to integrate with IBM FileNet Business Process Manager. The framework uses a well-defined rules interface that must be implemented to integrate with rules engines.
- **Process Analyzer**: Provides analysis capabilities to determine cycle times, find trends and bottlenecks, and generate reports and charts to analyze the processes deployed in the Process Engine system.
- **Process Simulator**: Allows customers to simulate processes and perform what-if scenarios with hypothetical or historical data.

- **Business Activity Monitor**: Provides monitoring of various aspects of the in-progress business processes in the context of other business applications in almost real-time.

### 3.2.3 Data tier

The data tier contains the BPM databases and directory service as shown in Figure 3-7.

![Data Tier Diagram](image)

*Figure 3-7  Data tier*

The data tier includes the following components:

- **Directory Service**: Provides authentication. The directory service functionality is accessed through Content Engine. Content Engine supports several LDAP providers and single-sign-on (SSO) functionality. (Note that in versions prior to Version 4.0, IBM FileNet Business Process Manager had direct access to the LDAP directory.)

- **Content Engine database**: Contains object stores. Each object store contains content and metadata about the content.

- **Process Engine database**: Contains configuration and running instances of processes.

- **Process Analyzer database**: Stores analytical data of executing processes.

- **Process Analyzer OLAP**: Contains aggregation of data stored in the Process Analyzer database for the purpose of enabling quick analytical charting and reporting.
3.3 Process Engine (PE) applications and other supporting applications

The Process Engine applications are implemented as applets and are hosted by IBM FileNet P8 Workplace in the Application Engine. With the exception of the Task Manager and Process Simulator applications, all the other applications use the Process Engine APIs. You can write similar applications with these APIs.

Figure 3-8 shows some of the Process Engine applications.

![Diagram of Process Engine applications]

The Process Engine applications include:

- **Process Designer**: Used to create, modify, and deploy processes.
- **Process Configuration Console**: Used to prepare and configure an IBM FileNet Business Process Manager system.
- **Process Administrator**: Used to query and modify executing processes.
- **Tracker (also known as Process Tracker)**: Used to look at the history of an executing process.
- **Process Simulation Designer**: Used to create simulation scenarios of business processes.
- **Process Simulation Console**: Used to manage and execute simulation scenarios.
- **Task Manager (also known as Process Task Manager)**: Used to manage some aspects of the IBM FileNet Business Process Manager system including the Component Integrator.
- **Step Processors**: Used to provide users with an interface to complete steps in a process. There are generic step processors provided by the IBM FileNet Business Process Manager environment. eForms and Lotus Forms can also be used as step processors. Specialized step processors can also be written using the Process Engine APIs.
3.3.1 Process Simulator system architecture

Process Simulator can be accessed from the Process Simulation Designer and the Process Simulation Console. The Designer is used to create simulation scenarios that can be saved and versioned in the Content Engine repository. The Console is used to manage the scenarios and to execute simulations. The result of a simulation execution is twofold: the simulation analytics are stored in a Process Analyzer database and the simulation object, containing animation information, is stored in the Content Engine for future animations.

Figure 3-9 shows the Process Simulator system architecture.

3.3.2 Process Analyzer system architecture

Process Analyzer has an Event Dispatcher that retrieves the events from the event logs in the Process Engine database, and a Publisher that processes the events. In addition, the Publisher retrieves the configuration data required to process the events. The configuration information includes the process definitions, queues, and users who are present in the event. At the end of the publishing interval, the Publisher updates the fact tables in the database.
The statistical data in the fact tables is used to build the OLAP cubes to provide the information required by reporting tools such as Microsoft Excel, IBM Cognos Analysis Studio, or IBM Cognos Reports.

Figure 3-10 shows the Process Analyzer system architecture.

![Process Analyzer system architecture diagram]

3.3.3 Business Activity Monitor system architecture

Business Activity Monitor (BAM) is based on IBM Cognos technology. It collects data from Process Analyzer through its built-in connection. It can also collect data from external applications through standard connections such as JMS, JDBC™, Web services, and EAI. The collected data can be either raw, as in the case of Process Analyzer, or context data. Context data is data used to correlate with raw data. For example, a raw data record containing an account number can be used to retrieve the account user name which corresponds to the context data.

The BAM component has an in-memory streaming capability, that is configured to generate in-memory views and OLAP cubes. Views of the data can be created, and rules can be applied. The rules can be used to calculate Key Performance Indicators (KPIs) and alerts.
The BAM Dashboard can be configured to display information stored in the views and cubes, and to create thresholds based on KPIs, configure alerts, and to perform actions if a threshold is violated. Figure 3-11 shows the BAM Dashboard system architecture.

![BAM Dashboard system architecture](image)

**Figure 3-11  Business Activity Monitor system architecture**

### 3.3.4 Business Process Framework system architecture

Business Process Framework (BPF) is a configurable framework for developing BPM Web client applications. The resulting Web client provides a feature-rich interface using HTML, Java Server Pages (JSP™), JavaScript pages, and other objects.

BPF applications are built based on a case paradigm. For example, an auto insurance claim can be considered a case. The generated application uses both Content and Process Engine APIs. The application user interface is metadata-driven and is built dynamically upon user request. The metadata describing the user interface layout is stored in the BPF configuration database and is managed via the BPF Explorer configuration tool.
3.4 Service Oriented Architecture (SOA)

IBM FileNet Business Process Manager provides a flexible architecture that organizations can leverage to implement business processes as Web services within enterprise architectures. IBM FileNet Business Process Manager solutions can connect processes with applications and systems using Web services and other standard integration methods. Organizations can increase the speed of development and create new business process management applications to support larger Service Orientated Architecture (SOA) initiatives.

IBM FileNet Business Process Manager processes can be published and consumed as Web services. The Process Engine leverages Web services technology to enable process designers to define message interactions internally between several IBM FileNet Business Process Manager business processes, as well as externally between IBM FileNet Business Process Manager and other systems.
Web services
Web services are the primary technology used in implementing Service Oriented Architecture. Web services have characteristics that make them the ideal building block for service oriented architecture. Web services are made up of a common set of protocols and standards. These define how Web services are published, discovered, and implemented in a standard format.

Figure 3-13 shows how a basic Web service is used. A client looks up a Web service from a UDDI registry and obtains its WSDL. It then invokes the Web service. In addition to UDDI, IBM FileNet Business Process Manager also supports the IBM WebSphere Service Registry and Repository (WSRR).

Simply having a Web service or a Web services API does not imply Service Oriented Architecture. It is the ability to orchestrate these Web services that is fundamental in building or integrating with Service Oriented Architecture. The ability to consume and publish Web services to create higher value services is an essential aspect of a Service Oriented Architecture. IBM FileNet Business Process Manager provides the facilities to participate in a Service Oriented Architecture.
Consuming Web services

An external Web service can be consumed by a business process by the use of an *Invoke* step. The *Invoke* step binds the process to the Web service. The ability to consume Web services in business processes allows the reuse of components from existing applications and this is key in building a Service Oriented Architecture.

Publishing Web services

IBM FileNet Business Process Manager provides constructs for publishing processes as Web services. A Web service can be exposed with the use of a *Receive* step. There can also be one or more *Reply steps* used to handle the Web services reply.

Publishing business processes as Web services allows business processes defined in IBM FileNet Business Process Manager to become reusable in other applications.

IBM FileNet Business Process Manager can inter-operate with enterprise architectures such as Enterprise Services Buses (ESB) and Web services management systems. IBM FileNet Business Process Manager can also help organizations automate and manage business processes and can be leveraged to build Service Oriented Architecture environments.

### 3.5 Process Engine logical view

At the core of IBM FileNet Business Process Manager is the *Process Engine (PE)*. From a logical view, the Process Engine can be seen as a collection of isolated regions¹, each one containing queues, rosters, and event logs. Processes are compiled and stored in an isolated region. Each process is defined as a collection of maps and the fields required to define the process behavior. Maps are a collection of steps organized into a directed graph required to complete a process. Figure 3-14 illustrates the Process Engine logical view.

---

¹ For an explanation of isolated regions, refer to “Isolated regions” on page 50.
**Process server**
A process server is a physical machine designated as a member of a Process Engine system. A Process Engine system is a farm of Process Engine servers, with one or more servers. A process server can only be a member of one Process Engine system. The process server runs the Process Engine software.

**Isolated regions**
An isolated region (as illustrated in Figure 3-14) is a logical subdivision of the Process Engine database. Isolated regions are often used to provide separate development, testing, and production environments which enable developers and testers the ability to efficiently utilize shared computing resources and provide the necessary isolation between these disparate activities. An isolated region consists of queues, rosters, and event logs.
Queues

A queue is a database structure that holds work items. There are three types of queues: user queues, work queues, and system queues:

- A user queue is a queue that holds work items waiting to be processed by a specific user.
- A work queue is a queue holding work items that can be completed by one of a number of users, rather than by a specific participant, or work items that can be completed by an automated process.
- System queues are queues holding work items that are undergoing or waiting for processing by the Process Engine server.

A roster is a database structure that stores the current location and other information about a work item. Process rosters provide the Process Engine software with an efficient way to locate a specific work item. An event log is a database structure that contains information about system-level events related to work item processing.

Process definition

A process definition (as illustrated in Figure 3-14 on page 50) describes a business process. It includes information such as fields and maps used in the process.

A map is a representation of a process that defines the sequences of steps needed to complete the business process. A step represents a stage in a process where a distinct, well-defined action is performed. On a process map, each step represents a specific activity or task in the business process described by the map. Routes between the steps indicate one or more possible processing paths. A field is a property that can be used in a process to hold data.

Process instance

A process instance corresponds to a single execution of a process definition. For example, the auto insurance claim for Account 120-021 is an instance of the claim process definition, where the Account field has the value 120-021. A process instance consists of work items.

A work item is a single unit of work composed of a collection of fields. Work items transverse the process map, moving the data required by the process from step to step as indicated by the map.
Process design with BPM

In this chapter we discuss the basic elements and concepts that are used in designing business processes with BPM. Our discussion is not meant to cover every topic, but to provide the foundation for designing a workflow process with BPM.

We cover the following topics:

- Basic concepts in process design:
  - Tasks, participants, and roles
  - Workflow properties (workflow groups and attachments)
  - Steps
  - Routing
  - Milestones
  - Deadlines
  - Task reassignment
  - Process voting
- BPM supported data types and system functions
- Inter-related processes
- Interaction with external systems
Notes:

- The example presented in the current chapter derives from the sample solution we described in Chapter 2, “Solution example” on page 23. We recommend reading that chapter before proceeding with this one.
- After absorbing the basic concepts from the current chapter, read Chapter 5, “Using Process Designer” on page 67 to learn how to use Process Designer to design your business processes.
- Then, in Chapter 6, “Implementing business processes: Case study” on page 127, we provide step-by-step instructions for actually implementing business processes with a case study.
4.1 Basic concepts in process design

IBM FileNet Business Process Manager comes with the Process Designer tool that enables you to design business processes for your business needs. Before using the tool, it is important to understand the key elements in a business process:

- Tasks, participants, and roles
- Steps
- Routing
- Deadlines
- Task reassignment
- Process voting

4.1.1 Tasks, participants, and roles

A task is a unit of work, inside or outside of an organization, which has a defined start and finish. Tasks are performed by people, departments, or systems, and are a part of business processes. A task can be something small and atomic, such as a person making a phone call; or it can be much more complex, such as renting a car.

People who perform tasks are participants for the defined tasks. In a business process, participants are assigned specific work that must be completed at one or more steps in the workflow. Participants can be people, a group of people, a department, or a system.

Participants who perform similar or related tasks can be grouped together in roles. A role is a collection of tasks assigned to, or expected of, a person or group. When a role is defined, it identifies all of the work that the participants will do, as well as the tasks that are associated with the step in the process.

In Chapter 2, “Solution example” on page 23, we described a sample solution for XYZ Corporation, where we automate their auto insurance claim approval process using IBM FileNet Business Process Manager. Let us revisit this example to better understand tasks, participants, and roles in a business process.

From the auto claim approval process example, we identify the following tasks:

- Interact with policy holders.
- Create and submit claim packet.
- Review submitted claims packets.
- Assign field agents to claims.
Gather supporting documents for claims including police reports, witness interviews, and photographs.
Review and process claims.
Send claim packets to adjustors.
Approve or rejects claims.
Issue funds for claim settlement to policy holders.
Classify claims received in the back office.

From the auto claim approval process example, we also identify the following participants:

- Agents
- Adjustors
- Field agents
- Accounting department
- Administration department
- Finance department
- Mail room

Roles tie tasks to participants. From our example, we identify the following roles in the claim approval process:

- Agents who interact with policy holders and create and submit claim packets
- Adjustors who review the claims packets and assign field agents to claims
- Field agents who gather supporting documents for claims
- Accounting department that processes claims and issues claim settlement funds to policy holders
- Administration department that performs administrative tasks such as classifying claims that are received in the back office, and sends initial claim packet to the adjustors
- Finance department that approves or rejects claims
- Mail room that classifies and processes claim packets that are received from the field agents

So in the XYZ Corporation example, employees and departments have tasks associated with them. In addition, internal or external systems might also have to perform tasks. We discuss that topic later.

As we stated before, a task is a unit of work, inside or outside of the organization, with a defined start and a finish, performed by people, departments, or systems. Part of designing a business process is to identify all the tasks, participants, and their roles in the process.
4.1.2 Workflow properties (workflow groups and attachments)

Workflow properties specify the subject and other general properties of a workflow. Some of the workflow properties include workflow groups, attachments, and data fields that can apply to any step in a workflow. The workflow properties apply to the entire workflow definition.

Workflow group
A workflow group represents a collection of one or more users or groups that can be assigned as workflow participants. The members of a workflow group typically perform a particular task or a set of tasks in a business process, for example, claims adjustors or supervisors.

For flexibility in defining a workflow, you can either assign one or more specific participants to a workflow group, or allow the participants to be assigned later, either as part of the launch process or at a particular step when the business process is running. This technique of using unspecified workflow groups in a workflow definition is useful when the participant for a particular task is likely to change each time the business process runs.

System workflow groups
BPM provides system workflow groups. F_Originator is a special workflow group that represents the name of the user who launched the business process. At launch time, the system assigns the name of the launch user to that workflow group.

All assigned members of the F_Trackers group become workflow trackers, with access to the Process Tracker application. Although it is not required, every workflow definition should have one or more users or groups assigned as trackers to monitor events and help resolve problems when a process is running.

Attachments
An attachment is a link to information that a participant uses to complete a step in a workflow. The specific item that an attachment links to is referred to as a target in the Process Engine documentation. The most common target is a document located in an object store. However, document arrays, stored searches, folders, URLs, or files located on a shared disk can also be targets.

You define attachments as part of the workflow definition properties, and then indicate which attachments will be used at each step.

An attachment can be designated as an initiating attachment. This designation means that the document used to launch the workflow will be assigned to the selected attachment. A workflow definition can have only one initiating attachment.
4.1.3 Steps

Steps represent specific business activities or system activities. In each step, you can assign participants (individuals, groups, and systems) to perform tasks.

A business process consists of a series of steps to be completed. How you actually create the specific steps (from the tasks and participants) in a business process depends on what is logical, necessary, and convenient for your business process application.

Using our claim approval example, we can design a matching step for each identified task in the claim approval process. If a task is complicated, you can break the task into multiple subtasks, thus multiple steps.

4.1.4 Routing

Routing specifies the sequence in which steps are executed in a business process. A route represents the path between two steps and defines the order in which steps will be processed. Aside from the first step and the last step in a business process, all other steps can have one to many inbound routes and one to many outbound routes.

Routing options and responses can be defined in a business process. The routing options are evaluated when the step is completed and they determine the next step in the business process. A list of available responses can be defined in each step so that the participants can select them as part of the work completed at the step. The response that a participant selects can also determine the next step in the business process. For example, if a participant approves a claim (that is, selects the approve response), the business process can take the approval route to complete the remainder of the work. If the participant rejects the claim, then a different route will be used.

There are three ways to incorporate routing options and responses in a business process:

- Single (linear) routing
- Conditional routing
- Parallel routing

Single routing describes a route in a business process where work can only continue along a single path. At most, one step in the workflow is active. There can be one or more outbound routes defined for a step, but only one step can be active at a time.
Conditional routing allows you to specify a condition on a route. A condition can be based on a response from a participant or evaluation of a data expression. If an expression evaluates to true, then that route is taken. An expression can also be a combination of response and data fields.

Multiple steps can simultaneously be split out to be processed in parallel during a business process. If steps do not depend on each other, then they can be processed in parallel. A step can have multiple outbound parallel routes, which results in multiple parallel steps. When all parallel steps are completed, the outbound routes from these steps are merged together in a collector step and the business process continues from there. The collector step is a special type of step that collects the parallel routes into one step. If data values vary from processing along different paths, then different merge options determine which values result in the collection step.

Let us see how different routes can be used in the claim approval example. We have the following two steps:

- **Step A**: An agent documents a loss due to an auto accident by a policy holder, and submits a claim packet to the back office.
- **Step B**: The mail room clerk logs and adds a timestamp on the claim packet in the back office.

The routing between these two steps is straightforward, single routing. There is a single active path from one step to the next. The task in the first step must be completed by the agent before the task in the second step can be completed by the mail room clerk.

Continuing with the same example, with some deviation for the purpose of explanation, we assume that after the mail room clerk logs the claim in the system (Step B), we have the following steps:

- **Step C**: A field agent needs to assess and report property damage.
- **Step D**: A medical reviewer needs to assess and report injuries resulting from the accident.
- **Step E**: When the property damage report and the injury report are filed, then the adjustor decides the claim amount.

Since Step C and D are two steps that are independent of each other, we can design them to work in parallel. Thus, we can use parallel outbound routing from Step B, to go to Step C and Step D in parallel. Then we can have a collector step to ensure that both finish before it routes to Step E.

Routes ensure that work is sent to the right participants to be processed. The combination of defined steps and routes becomes a template that is used each time the business process runs.
4.1.5 Milestones

Milestones are defined as key notification points in a business process. These are points at which participants need to receive a message describing an important task or tasks that are pending or completed.

Each milestone can be used in one or more steps. The status message can be either specific text, or it can include the values of one or more data fields. When the workflow process reaches a milestone, the specified message can be displayed for participants and users who launched the workflow.

For our claim approval example, we do not use any milestones.

4.1.6 Deadlines

*Deadlines* are time-based constraints that can be applied to steps in a business process. A step with a deadline implies that the participant must complete the step within a specified amount of time. The specified time (the deadline) is relative to the time that the step starts (when the participant receives the work packet).

A participant that is routed a task with a deadline can receive a reminder of the pending deadline via e-mail. When a deadline is reached, e-mail messages are sent to the step participants (with the uncompleted step) and the workflow trackers. Users can update their e-mail notification preferences to turn off different types of e-mail notification. When a deadline expires, it can also be set to trigger escalation such that the work packet is routed to a special escalation step to be performed by a supervisor or some other participants. Using a deadline provides added control and an escalation mechanism for a business process.

Using our claim approval example, we have the following step that might require implementation of a deadline:

- Step: The adjustor assigns a field agent.

The adjustor assigns a field agent. If this task does not happen in a timely fashion, it impacts the entire claim approval process. By adding a deadline on this step, we can add some time management to the step to ensure that the adjustor is not delaying the claim approval process. We can also set the adjustor's supervisor as the tracker for the workflow (by adding the supervisor to the F_Trackers group for the workflow). When a deadline or a reminder is reached, both the adjustor and the supervisor are notified. Again, receivers of the reminder have options to turn off different types of e-mail notifications.
When a deadline expires on a step and a submap is specified, the work is routed to the first step in the submap.

Of course, we can also automate the field agent assigning step such that the adjustor does not manually assign a field agent to the claim; rather, the system automatically assigns a field agent based on the place of the incident. In this scenario, a deadline is not required.

The task of gathering supporting documents for a claim might be a good step in which to apply a deadline. Although a participant might not always be able to complete a task in a timely manner due to factors outside of their control, deadline mechanisms (including reminder e-mails and automatic escalations) provide effective time management and control to ensure more timely completion of a step.

### 4.1.7 Task reassignment

For flexibility and control in a business process, you can design a step to enable the original participant to dynamically reassign the tasks to other participants in the step. Specifically, a participant can assign another participant to take over the task or delegate the work to another participant. In the IBM FileNet Business Process Manager context, when reassigning tasks within a step, two options are available, delegation and abdication:

- When using the *delegation* option, a participant reassigns the task to another participant. After the task is complete, the work packet comes back to the original participant for review before the work packet moves on to the next step in the business process.

- When using the *abdication* option, a participant reassigns the task to another participant. After the task is complete, the work packet is directly sent to the next step in the business process.

Reassignment can be disabled on any user step in the workflow.

### 4.1.8 Process voting

There are instances when the system requires input from multiple participants to determine the next step in the business process. This calls for *process voting*. When a process voting is implemented, each participant will vote and the results of the voting will determine the next step in the business process.

You can implement the routing decision based on one of the following conditions:

- When all participants select the same response (for example, when all approve)
As long as one participant responds a certain way (for example: when at least one person rejects).

- When none of the participants select a particular response (for example: no one rejects).

- Simple majority, when most participants select a certain response (for example: most participants approve rather than reject).

Let us see how this can be applied to the XYZ Corporation’s claim approval business process example (we will deviate somewhat from the actual example for the purpose of explanation).

We assume that, in the claim approval process, there is a step when the claim approval decision task is escalated to a supervisor if the claim is determined to be a high risk (for example, the claim amount is extremely high). According to the business rules in place, multiple risk managers make the approval decision. The supervisor thus routes the decision task to the risk managers who are experienced in dealing with assessing risks. The risk managers review the claim and then each either approves or rejects the claim. Per the business rule in the example, as long as the majority of the risk managers approve, the claim is deemed approved. In this case, we would implement the simple majority condition for processing voting.

The process voting function allows organizations to implement collaborative decision making while still keeping within a structured business process.

### 4.2 BPM supported data types and system functions

IBM FileNet Business Process Manager (BPM) supports a set of data types and provides a set of powerful system functions that you can use for your process design.

#### 4.2.1 Supported data types

BPM supports literals and variables of the following simple and multiple (arrays) data types:

- Boolean
- Float
- Integer
- String
- Time
4.2.2 System functions

IBM FileNet Business Process Manager provides a set of functions to use in any expression. Some example functions include data type conversion functions, system interrogation functions, general functions, and time functions.

Data type conversion functions
BPM supports data type conversion. Conversion must be explicit in string variables. The data conversion functions include:

- `convert(source_exp, type_name)`
  Converts an expression from one supported data type to another.
- `numbertostring(num_exp{, num_mask}opt)`
  Converts an integer or float to a string, using a specified mask.
- `timetostring(time_exp{, time_mask}opt)`
  Converts a time to a string, using a specified mask.

System interrogation functions
System interrogation functions include the following function:

- `systemtime()`
  Returns current server time.

General functions
General system functions include the following functions:

- `if(bool_expr, expr2, expr3)`
  Conditional expression returns expr2 if true or expr3 if false.
- `sizeof(array_id)`
  Returns the size of the array.

Time functions
Time functions include the following function:

- `adddays()`
  Adds a specified number of days. Functions are also available for minutes, hours, and so forth.

Queue operations
An operation is a function within a step that performs a specific task associated with a particular queue. When an operation and its parameters are being defined,
the process designer, system administrator, and application developer typically work together to determine its names and usage.

Operations can be created in work queues, component queues, and user queues. For information on work queues, component queues, and user queues, refer to “Queues” on page 51.

4.3 Inter-related processes

The business processes in an organization can be very complex. There are occasions when one business process relies on another business process to complete its work. The other process might even start this business process. The concept of inter-related processes involves the notion of one business process launching another to carry out additional processing.

When designing business processes, understand which other business processes are already available that you can incorporate and re-use. In addition, you can decompose your business process into multiple logically separate processes that can be shared and used by other processes.

Referring back to the claim approval example, there are areas where we can leverage inter-related processes. Consider the following steps:

- **Step A:** An agent documents a loss due to an auto accident by a policy holder, and submits a claim packet to the back office.

- **Step A2 (new step):** The agent performs an initial review on the claim.

- **Step A3 (new step):** If the claimant claims that the insured car is no longer in operation due to the accident and the policy allows for a rental car in this type of situation, then the agent calls another entity within the company to set up a rental car for the claimant, gets the rental car confirmation number back from the entity, and passes that confirmation information to the claimant.
Step B: The mail room clerk logs and adds a timestamp on the claim packet in the back office.

In Step A3, another group is involved in making a car reservation for the claimant or any individual within the company. This is an independent business process performed by the group. If it already exists for the company, then the claim approval process can call the car rental process to perform the task. The agent, involved in the claim approval process, does not have to understand the actual process of reserving the rental car. All it requires is for the agent to pass the necessary claimant information and retrieve the confirmation number later.

If, at the design time of the claim approval process, the car reservation process is not created within the system, you can create this separate process and call it in the claim approval process.

4.4 Interaction with external systems

IBM FileNet Business Process Manager provides an extensive collection of tools for building and customizing solutions and integrating with external systems and processes. There are several options for business processes to interact with external systems: Component Integrator, Web services, and EAI connectors.

For the rental car reservation process that is mentioned in 4.3, “Inter-related processes” on page 64, the reservation process can be a process performed by the rental car company that is transparent to the agent. If such an automated process exists in the rental car company, then the claim approval process can interact directly with the external process to make the appropriate rental car reservation. All the process has to do is to have the agent to approve the rental car, after which it requests the external reservation process to reserve the car and retrieves the reservation number automatically.

Which technology to use (either Component Integrator, Web services, or the EAI connector) depends on how the external system works and what is available for the claim approval process to interact with the external system.

For more information regarding direct interaction with the external systems, refer to Chapter 7, “Integration with external systems and services with BPM” on page 257.
Using Process Designer

In this chapter we show you how to use the Process Designer tool to create a business process definition.

We cover the following topics:

- Process Designer user interface overview
- Creating workflow definition overview
- Setting up workflow properties
- Creating steps in a workflow map
- Creating routes with Process Designer
- Advanced topics:
  - Workflow inheritance
  - Default system submaps
  - Expression Builder

For process design concepts, refer to the previous chapter. For creating a case study business process using Process Designer with step-by-step instructions, refer to the next chapter.

Note: The online ecm_help provides much detailed information on using Process Designer to create workflow processes. For your convenience and the completeness of this book, we extract some content from the online help in this chapter. For more detailed information regarding using Process Designer, refer to the online help.
5.1 Process Designer user interface overview

Process Designer is a graphical tool that you use to create and modify business process definitions. Process Designer is one of several tools provided with IBM FileNet Business Process Manager.

Using Process Designer, you can create the specifications for each step in a business process, including which participant processes the work for the step; what interface application (also known as a step processor) the participant uses to perform a task; what attachments, if any, are required for the process; what data is necessary to view, add, or edit; what responses the participant can choose; and what routes to take within a business process. The electronic representation of the business process designed in Process Designer is called a workflow definition (also known as a process definition).

Note: In IBM FileNet Business Process Manager, a business process is more than just a simple workflow. However, for convenience when writing for this book and as well as consistency with the wording used in the Process Designer application, we use workflow interchangeably with business process from this point on.

Process Designer is launched from the IBM FileNet Workplace XT application by selecting Tools → Advanced Tools → Process Designer. See Figure 5-1.

Figure 5-1  Launching Process Designer from Workplace XT
Alternatively, you can launch it from IBM FileNet Workplace by selecting **Author → Advanced Tools → Process Designer** from the navigation bar on the left hand pane. See Figure 5-2.

![Figure 5-2](image)

**Figure 5-2  Launching Process Designer from Workplace**

**Note:** After launching the Process Designer application, *do not* close the IBM FileNet Workplace Internet Explorer® window. Closing the Internet Explorer window will result in the closing of the Process Designer application.

Figure 5-3 shows the Process Designer user interface. The window consists of the Map pane, Properties pane, and Step Palette pane. There is also a set of toolbar icons on top of the Map pane.
5.1.1 Map pane

A business process consists of one or more workflow maps. You create the maps in the Map pane of Process Designer. The Map pane is displayed on the top left of the Process Designer window. See Figure 5-3. In this pane, you define steps and routes. There is always one main map and optionally multiple submaps.

**Note:** This book is written based on Version 4.0 of IBM FileNet Business Process Manager. For release 4.5 or later, some of the user interface, including menus and toolbar, might be changed.
5.1.2 Properties pane

The Properties pane is displayed on the top right of the Process Designer window. The Properties pane displays the properties for the selected step or route in the Map pane. When you create an object in the map (a step or route), you specify the properties for that object in the Properties pane. The configurable properties vary depending on the type of step or route that is selected. For example, if the selected object is a General step, you can specify a step destination, required parameters, milestones, deadlines, and other options that are specific to the General step. If the selected object is a route, you can specify participant responses, field values, and routing conditions.

Figure 5-4 shows the Properties pane for the LaunchStep. For this step, you can provide the user with instructions for how to complete the step in the General tab, select parameters to read and write in the Parameters tab, modify parameter values in the Assignment tab, and specify routing information in the Routing tab.

![Properties pane](image)

Figure 5-4 Properties pane

We discuss the various settings in the Properties pane for different steps and routing in later sections.

5.1.3 Step Palettes pane

The Step Palettes pane at the bottom left of the Process Designer interface contains collections of predefined steps that represent different types of activities in a workflow.
Step Palettes include the following palettes:

- BPM Palette
- CheckPoint Palette
- General System Palette
- Timer Palette
- Web Services Palette
- My Palette

Figure 5-5 shows the Step Palette drop-down menu in Process Designer.

![Step Palettes in Process Designer](image)

**BPM Palette**

The BPM Palette (Figure 5-6) contains common types of steps you use for a workflow.

The BPM Palette comes with the following types of steps:

- General step
- Submap step
- System step
- Component step
General steps are used to handle basic workflow activities (tasks) that are performed by an individual participant or a group of participants. System steps are used to call built-in system functions. Component steps are used to call operations from an external application.

To simplify a workflow map, you can break a workflow into one main map and several smaller submaps. To call the submap from a main map, you insert a Submap step and specify which submap to call at the step. A submap can call other submaps.

**General System Palette**

The General System Palette (Figure 5-7) contains the steps that call system functions.

The General System Palette comes with the following types of steps:

- Assign
- Create
- DbExecute
- Delay
- Log
- Return
- TerminateBranch
- TerminateProcess
- WaitForCondition

All of these system functions can be called directly in the system step available from the BPM Palette. The difference is visual. By using the system step from the BPM Palette, you do not know what system function the step calls in the map unless you look into the step properties (or you can provide a very descriptive name for the step that shows on the map). If you use the step from the General System Palette, you can tell what the step does by its name.
The advantage of using the system step in the BPM Palette is that it enables you to call multiple system functions, one after another within the same step. Thus, one system step can perform, for example, DbExecute(), Create(), and then TerminateBranch(), and they all appear as one single step in the workflow map. This simplifies the visual appearance of the entire workflow map.

**Timer Palette**

The Timer Palette (Figure 5-8) contains steps that allow timer functions to interact with steps in the workflow definition. The timer functions are used to specify processing time limitations for steps in the workflow. This is different than the deadline function we discussed earlier in Chapter 1, “Overview” on page 1. A deadline applies to a specific step. A timer usually applies to a series of steps.

The Timer Palette includes the following types of steps:

- BeginTimer
- SuspendTimer
- ResumeTimer
- EndTimer
- EndAllTimers

![Figure 5-8 Timer Palette](image)

**CheckPoint Palette**

The CheckPoint Palette (Figure 5-9) contains steps that allow the checkpoint system functions to interact with steps in the workflow definition. The checkpoint functions provide mechanisms that allow data field values of work items to be rolled back to an earlier state. Checkpoint functions also make it possible to resume processing for work items at a previous point in the workflow.

The CheckPoint Palette includes the following types of steps:

- BeginCheckpoint
- RollbackCheckpoint
- EndCheckpoint

![Figure 5-9 CheckPoint Palette](image)
Web Services Palette
The Web Services Palette (Figure 5-10) contains the steps that allow the workflow definition to interact with Web services system functions. The steps specify where in the workflow a Web service can be invoked or where in the workflow a step can receive a reply from a Web service.

The Web Services Palette includes the following types of steps:

- Invoke
- Receive
- Reply

![Figure 5-10  Web Services Palette](image)

My Palette
This palette allows you to create your own collection of step types you use most often in designing your workflow. For example, if you create a custom step that is reusable in another workflow, you can add the custom step to My Palette (drag it from the workflow map in the Map pane) and then use it later for a different map. You can treat My Palette as a temporary holding place for commonly used or reusable steps.

We cover what you can set for each step in later sections. For more detailed information about what each step does, refer to ecm_help.

5.1.4 Toolbar icons

Below the menu in the Process Designer, there is a list of icons (Figure 5-11). They are shortcuts to access various functions of Process Designer. Here, we list and group them in the order they appear in Figure 5-11:

- New Workflow
- Open
- Save

--------------------------
- FileNet Open/Checkout
- FileNet Add New
- FileNet Checkin
- FileNet Cancel Checkout
- FileNet Save

--------------------------
5.2 Creating workflow definition overview

Using Process Designer, you create a workflow definition (also known as a process definition). As mentioned earlier, a workflow definition describes all activities required to complete the business process.

A workflow definition contains workflow properties and workflow maps. Workflow properties contain properties global to the entire workflow. Sample workflow properties include all data fields that can be used in a workflow, all possible attachment types that a workflow can have, and all participants that can process work items in the workflow. Workflow maps include steps and routes.

For each step, you specify the actual participants that will process work items at this step, as well as the data fields available at this step for reading and writing by systems or participants. There are other configurations that you can set up in each step, depending on the type of step. Figure 5-12 illustrates the elements that comprise a workflow definition.
To create a workflow definition, you must set up workflow properties and create the appropriate workflow maps.

5.3 Setting up workflow properties

Each workflow has a set of workflow properties. It is different than the properties shown in the Properties pane of the Process Designer window because the workflow properties are global to the entire workflow, whereas the Properties pane only shows the properties of the selected object on the workflow map.

To set or view the workflow properties, click the Workflow Properties icon from the toolbar in the Process Designer, or select View → Workflow Properties from the menu. See Figure 5-13.
The following tabs are shown on the Workflow Properties dialog box:

- General tab
- Advanced tab (roster, event log, condition identifier, and e-mail notification)
- Data Fields tab
- Attachments tab
- Workflow Groups tab
- Maps tab
- Milestones tab
- Web Services tab

**General tab**

Figure 5-14 shows the General tab in the Workflow Properties dialog box.

![Workflow Properties - General tab](image)

The General tab allows you to specify general information for the workflow, including base workflow, name, subject, description, and deadline.

From the General tab, set up the following fields:

- **Base Workflow**

  Select a base workflow that the new workflow definition is based on. You can base your new workflow definition on an existing workflow (the base workflow) to inherit the properties and submaps defined in the base workflow.
To create a workflow definition based on default settings, leave the default WorkObjectEX as the base workflow. To create a workflow definition based on some previously defined workflow, click the Modify icon and select an appropriate base workflow.

Refer to 5.6.1, “Workflow inheritance” on page 117 for more information on how workflow inheritance works.

 Workflow Name
Enter a name for the workflow definition. It is displayed in the list of workflows when the workflow definition is transferred to the Process Engine. The name of a workflow definition must conform to the following rules:

- The name must begin with an alphabetic character. It can contain letters, digits, underscores, or spaces.
- The maximum length for the name is 249 characters.
- The name must not contain F_ or two tilde (~~) characters as the first two characters.
- The name must be unique among the workflow definitions in an isolated region.

 Subject
Enter the title that describes the workflow. This is what participants see in their inbox when the workflow is running and there are work items for them to process from this workflow.

 Description
Enter a description for the workflow. This is an optional field.

 Deadline
Specify a time limit for the workflow to be completed in minutes, hours, days, or weeks. This time starts when the workflow launches. You can also set a reminder notification prior to the deadline. This is an optional field.

If the e-mail notification is set, then the workflow tracker receives the reminder notification. To enable participants to be notified of the workflow deadlines, you set a deadline in individual steps of a workflow.

Advanced tab
Figure 5-15 shows the Advanced tab in the Workflow Properties dialog box.
The Advanced tab allows you to specify advanced features for the workflow, including roster, event log, condition identifier, and e-mail notification.

Set up the following fields in the tab:

- **Roster**
  Select a roster for the workflow. There is always a default roster selected.
  Roster is a database table that stores the current location of the workflow (which step it is at, and which working queue the work item is in) and other information about workflows. Workflow rosters provide the Process Engine with an efficient way to locate a specific workflow.

- **Event log**
  Select an event log. There is always a default event log selected.
  An event log is a database table that contains information about system-level events related to work item processing.

- **Condition identifier**
  Enter an expression that identifies a specific work item at runtime. This value is used by the WaitForCondition system function to identify a specific waited-for work item.
  When a work item has to wait for another item to complete or meet a certain condition, you can use the WaitForCondition system function. For example, you have a main workflow map and multiple submaps. In the main workflow map, it requires the completion of a submap to continue the main map’s process. Setting a condition identifier enables the submap to communicate with the main map about its completion.
Enable Email Notification

Enable or disable e-mail notification for the current workflow definition.

Data Fields tab

Figure 5-16 shows the Data Fields tab in the Workflow Properties dialog box.

![Workflow Properties - Data Fields tab](image)

The Data Fields tab allows you to define all data fields that can be used in workflow steps. When you define each step, you specify which data fields are parameters for that step, and what access permission the participants have to modify the values. Enter the following information for each data field:

- **Name**

  Enter the name for the data field.

- **Type**

  Select one of the data types for the field: Boolean, Float, Integer, String, Time. If the field has multiple values, then select the array version of the type indicated by brackets [ ].

- **Merge Type**

  Specify how values will be merged when necessary.

  A step in a workflow can have multiple outgoing routes if multiple tasks (following this step) can be performed in parallel. When they merge back to one step, this field specifies how values from multiple sources of the same data type are to be merged into one. The default for Merge Type is Override.

- **Expression**

  Specify an initial value for the field. String fields can be blank, but all other types require an initial value. The value must match the data type and meet the requirements for expressions. From our example in Figure 5-16, we use systemtime() for several date fields.
Introducing IBM FileNet Business Process Manager

Attachments tab
Figure 5-17 shows the Attachments tab in the Workflow Properties dialog box.

![Workflow Properties - Attachments tab](image)

The Attachment tab allows you to specify attachments that participants use in workflow steps. Attachments include documents and URLs. For each attachment, specify the following information:

- **Name**
  Enter the name for the new attachment.

- **Array**
  Specify whether there are multiple attachments. If the Array checkbox is selected, it represents multiple attachments. For our example, as shown in Figure 5-17, we select the Array checkbox for SupportingDocs. This is because we expect the workflow to have multiple supporting documents.

- **Value**
  Specify a value for the attachment. If it is a file or folder in an object store or library, you can double-click the field and select one from a list. Alternatively, you can leave it blank. This allows the value to be assigned at the workflow launch time or by a participant during workflow processing.

- **Description**
  Provide a description for the attachment. This is an optional field.
Workflow Groups tab

Figure 5-18 shows the Workflow Groups tab in the Workflow Properties dialog box.

![Workflow Groups tab](image)

Figure 5-18  Workflow Properties - Workflow Groups tab

The Workflow Groups tab allows you to specify an individual user or a group of users who can be assigned as participants in the workflow. For each workflow group, specify the following information:

- **Name**
  Enter a name for the workflow group.

- **Participants**
  Select users for the workflow group. Click **Modify** to select users.
  If a step is processed by a workflow group, then the participant or the group must be defined in the workflow group here before the step can use it in its setup.

- **Descriptions**
  Provide a description for the workflow group. This is an optional field.

Maps tab

Figure 5-19 shows the Maps tab in the Workflow Properties dialog box.
Introducing IBM FileNet Business Process Manager

Figure 5-19  Workflow Properties - Maps tab

The Maps tab allows you to specify the main workflow map and the submaps that you use in your workflow. You can create more flexible and modular workflow definitions by creating submaps. In addition, you can redefine the behavior of maps that are inherited from the base workflow. For example, by default, you get Terminate and Malfunction submaps. Using submaps and the ability to redefine inherited maps enable you to create custom actions and processing for your workflow process.

For each map, define the following information:

- **Name**
  
  Enter a name for the main workflow map or submap. Indicate whether it is the main workflow map that launches the workflow. The main map that drives the workflow process is denoted with a special pencil next to the down-arrow icon.

- **Description**
  
  Provide a description of the map. This is an optional field.

If there is a down-arrow icon next to the map, it means that the map is inherited from another map. If you create a new map from scratch, there will not be an icon next to the map.

Figure 5-20 shows the Maps tab with a newly created submap called Claim Verification. Clicking the Map Usage icon on the upper right corner reveals the map that calls the Claim Verification submap. See Figure 5-21.
**Milestones tab**

Figure 5-22 shows the Milestones tab in the Workflow Properties dialog box.
The Milestones tab allows you to define milestones in a workflow. Milestones help track the progress of a workflow process. When the workflow reaches a milestone, a milestone message is written to a log file. Depending on the milestone message level, the message is displayed for workflow participants, trackers, and users who launched the workflow. The message can be text or it can include the value of one or more data fields. The maximum display length for a milestone is 250 characters. For each milestone, define the following information:

- **Name**
  Enter a name for the milestone.

- **Message level**
  Specify a message level for the milestone from 1 to 99.

- **Message**
  The milestone message that will be recorded in the log when this milestone is reached. The maximum runtime display length is 250 characters. Characters beyond 250 are truncated.

### Web Services tab

Figure 5-23 shows the Web Services tab in the Workflow Properties dialog box.

![Workflow Properties - Web Services tab](image)

Figure 5-23  Workflow Properties - Web Services tab

The Web Services tab allows you to specify Web Services to be used in the workflow. Set the following information for the Web Services:

- **General tab**
  Enter general settings for invoking or providing Web services.
– Incoming Web Services Attachment Folder

If the workflow uses Web services operations that include incoming attachments, specify a folder in the object store where these files will be stored.

– Verify messages at runtime

When an XML message is received either as incoming data for a Receive system instruction or as a reply message from a Reply system instruction, specify whether to use the appropriate schema to validate the message at runtime as it is received.

– Finalize existing Web services operations

Select this option to prevent changes in the input and output parameters of existing operations in Receive and Reply functions in this workflow definition.

▶ Partner Links tab

Specify the Web services partner links that are used in this workflow. Partner links consist of a name and the WSDL URL for Invoke functions, or a name and port definition for Receive functions. See Figure 5-24. These are required before you can create Invoke, Receive, or Reply system functions.

▶ XML Schema tab

Specify an XML schema for complex data in a Receive or Reply function. See Figure 5-25.
XML Data Fields tab

Used for Web services operations that use XML mode instead of parameter mode. It can also be used to specify XML fields for uses such as messages. See Figure 5-26.

We discuss Web services and how to set one up in later chapters of the book.

5.4 Creating steps in a workflow map

A workflow definition consists of workflow properties and workflow maps. A workflow map consists of two or more steps. To create a workflow map, you define the steps that are involved in a workflow process and the routes that specify the sequence of the steps.
Each step in a workflow represents a unique task or activity to be completed by a participant, a group of participants, or a system. Process Designer contains many predefined steps (or to be more precise, step templates) to perform different types of activities. These steps are contained in the Step Palettes and can be added to workflow maps as needed. Recall that the BPM Palette contains the most common steps used in the workflow:

► General step
► Submap step
► System step
► Component step

There are two other steps that can be found on a workflow map:

► Launch step (the default first step in a main workflow map)
► Start step (the default first step in any submap)

Figure 5-27 illustrates what some of the steps look like in a workflow map. Notice that for each type of step, there is a different icon symbol representation.

![Main workflow map with different types of steps](image)
Every workflow map has a default launch step. This is the first step when the workflow launches. The submap step, shown in Figure 5-27, with the down arrow icon symbol (it looks like it is drilling down to the next level) calls a submap to perform activities. The component step with the electric plug icon symbol (it looks like it is plugging into something) calls external systems or services to perform activities. The general steps with the people icon symbols (they look like they are something to be processed by humans) require participants to perform activities at the step. There are other icon symbols for the general steps, depending on who or what processes the activities at the step. We discuss configuring these steps in more detail later in this section.

**Adding a step**
To add any step to a workflow map:

1. Use the Step Palette drop-down menu and select the appropriate palette. Usually, this is the BPM Palette.
2. Select the step icon from the palette and drag it to the Map pane.

The user can also add a step to the map using the map's context menu.

**Configuring a step**
To configure a step that you added to a workflow map:

1. Select the step icon in the workflow map.
2. Edit its properties in the Properties pane.

Depending on the type of the step, you can configure different settings for it. We cover that in more detail later in this section.

**Deleting a step**
To delete a step from a workflow map:

1. Right-click the step icon in the workflow map.
2. Select **Delete** from the context menu.

You cannot delete the default steps (launch step and start step) that are always included in a main workflow map and its submaps.
5.4.1 General step

A general step is the most frequently used type of step in a workflow map. At each general step, you assign work (a task) for a participant to complete. When the workflow moves to this step (you can also think of when the work items are moved to this step) and a participant is assigned the task, the task shows up in the participant’s inbox.

You can assign a task to a work queue in the general step. Anyone who has access to the work queue can perform the task. Assigning work to a work queue offers more flexibility than assigning a task to an individual. How you assign tasks depends on your business practice.

Many of the configuration settings for the general step apply to other types of steps. Use the setting discussed here as a reference for configuring similar properties for other types of steps.

To configure a step, provide a name in the Name field of the Properties pane. We recommend providing a meaningful, descriptive name. The name appears in the workflow map below each step icon and it provides information about what the step does.

Other properties of a general step can be configured through the following tabs:

- General tab
- Deadline tab
- Parameters tab
- Assignments tab
- Routing tab

If you configure your workflow to use a rules engine, there is also a Rules tab. It is beyond the scope of this book to cover rules engines. We do not discuss the configuration of rules engines in this book.

**General tab**

Figure 5-28 shows the General tab in the Properties pane for a general step.
The General tab allows you to configure general information about the step, such as the participant’s instructions for the step, the step destination, and the step processor.

From the General tab, set up the following fields:

- **Instruction**
  Enter the instruction that displays on the user interface when the participant processes a task at this step. For our example in Figure 5-28, we have an Adjustor Review step. When the adjustor processes work at this step, we provide the following instruction as shown in Figure 5-28:
  
  Please Review the attached claim and related documents and Approve or Reject the claim.

- **Step Destination**
  Select whether the step is assigned to a participant or a work queue.
  As mentioned earlier, depending on your business practice, you can assign work to a participant or to a work queue where anyone who has access to the queue can process the work.
To assign work to a work queue, select **Work Queue**, and select an existing queue from the drop-down list. If an automated process has access to the queue, the automated process can also process work at this step.

To assign work to participant, select **Participants**. To add the participants, click the **Modify** icon (denoted as a pencil symbol) on the upper right corner of the Participants entry box. The Participant Selection dialog box appears. You can select workflow groups, users, or groups for the participant. If you do not see the workflow groups that you want in the selection list, make sure to add them in the workflow properties first.

Figure 5-29 shows three icons. When the step destination is assigned to a work queue, participant, or unassigned, it has a different icon representation for ease of reading the workflow map.

> **Ignore Missing Participants**

If Participant is selected for the step destination, specify whether to ignore missing participants if they are missing from a workflow group. If this flag is not set and there is no valid participant who can process the work, the workflow will stop at this step and produce an error. For our example in Figure 5-28, we want the work to stop if there is a missing participant; therefore, we do not select the check box.

> **Participant Privileges**

If Participant is selected for the step destination, specify whether the participant can reassign work to someone else, view the status of the workflow (through the Tracker tool), or view the workflow history. For our example in Figure 5-28, the adjustor has all the privileges.
Step Processor

Step processor is the user interface that a participant uses when processing a task at a step. Select from a list of the step processors that can be used for the participant to process the work.

Deadline tab

Figure 5-30 shows the Deadline tab in the Properties pane for a general step.

![Properties pane - Deadline tab](image)

For time management, you can set a deadline for when the task should be completed at this step. Reminders can also be set to notify participants of an approaching deadline. The deadline set in the workflow properties applies to the entire workflow process. The deadline set here applies to this step only.
As mentioned in the previous chapter, you cannot force a participant to finish the task before the deadline. However, setting a deadline helps to provide some time management. If the participant does not finish the task on time, an icon indicating the expired deadline will display next to the task that the participant has to complete in the inbox. Also, you can set up escalation points here to notify a supervisor or another control group to manage the work.

From the Deadline tab, set up the following fields:

- **Complete within**
  
  Specify when the task should be completed at this step. Use the first option to specify a fixed number of time period for the deadline. Use the second option to enter an expression that determines when the deadline should be.

  For our example, in the Adjustor Review step, we set a deadline of 1 week. This means that the adjustor has one week to complete the task from the time the task is assigned to the adjustor at this step.

- **Send Reminder**
  
  Specify whether there should be a reminder sent to the participant within a certain period before the deadline.

  For our example in Figure 5-30, we configure a reminder to be sent to the adjustor 2 days before the task’s deadline.

- **Deadline Submap**
  
  If the deadline expires, you can activate a deadline submap to handle the situation.

  The deadline submap can be any submap you design to handle the situation when the deadline expires.

Setting up a deadline for a step is optional. You can choose not to set deadlines, or selectively set up different fields. For example, you can set up a Complete within field, without setting up the Send Reminder and Deadline Submap fields.
Parameters tab

Figure 5-31 shows the Parameters tab in the Properties pane for a general step.

You can specify the operations and parameters that can be used for read and write at this step. Parameters include data fields, attachments, workflow groups, and XML fields. You can use the parameter values to help you perform the task at this step or update their values as part of the task. There is a list of available parameters for you to choose from. If the one you want does not appear in the Available Parameters list, go back to the workflow properties and add it there. Also, the checking of any of the four icons above the list determines what parameters appear on the available list.

From the Parameters tab, set up the following fields:

- **Operation**
  
  If the work queue assigned as the destination step has operations defined, you can select the operation to be executed here.
Select Parameters
Select the parameters that you want to be used at this step. For our example, some of the parameters are AccidentDescription, Agency, AgencyPhone, AgentName, CaseFolder, and ClaimID. This information is important for the adjustor to use at this step.

Access Rights
Indicate whether the participants have read, write, or both read and write permission for the selected parameters. For our example in Figure 5-31, the adjustors can read the CaseFolder and ClaimID, but they cannot update these fields. They can however, read and write (update) many other fields.

Prompt
Enter a descriptive prompt for participants when they work with a parameter. For example, when the adjustor is working with the AccidentDescription field, the prompt for the adjustor is:

Description of the accident

Assignments tab
Figure 5-31 shows the Assignments tab in the Properties pane for a general step.

![Properties pane - Assignments tab](image-url)
In the Assignments tab, you can set the milestone information, assign values to fields using expressions, and set up attributes before the execution of the step (in the Before Execution subtab) or after the step is completed (in the After Completion subtab).

From the Assignments tab, set up the following fields:

- **Milestone**
  
  Assign a milestone before or after the step is completed.

- **Field Assignments**
  
  Assign a value to a field. You can enter an expression as the value for the field.

- **Attributes**
  
  Specify name, type, and value for the attributes. Attributes exist only in the step for which they are set. They are like the temporary parameters specific for the step.

Setting up assignments for a step is optional. You can choose not to set them or selectively set up different fields. For example, you can set up Milestone without setting up Field Assignments and Attributes information.

**Routing tab**

Figure 5-31 shows the Routing tab in the Properties pane for a general step.

![Properties pane - Routing tab](image)
In the Routing tab, you configure routing-related settings such as routing responses, as well as incoming and outgoing routing settings.

From the Routing tab, set up the following fields:

- **Responses**
  Specify a list of possible responses participants can select at this step. The response determines the routing from the current step to the next step. For our example in Figure 5-33, we show the Adjustor Review step’s Routing tab. At this step, the adjustor can either deny the claim, approve the claim, or escalate to a supervisor for review. We therefore enter three responses for the step: Approve, Escalate, and Reject. You can rearrange the order of the responses by using the upper and lower arrows on the upper right corner of the Responses entry.

- **Incoming Routing Information**
  If you used parallel routing earlier, at some point, you must have a collector step to combine the parallel routes into one place. If the current step is the collector step, check the **Collector Step** option. Otherwise, leave it unchecked.

- **Outgoing Routing Information**
  If there are multiple routes leaving from this step to other steps, you specify how the route is determined in this field. You can select one of the following options:
    - All true conditions: All specified conditions must be true before the route can be taken.
    - First true conditions: As soon as one condition becomes true, the route is taken.

There are three options listed under the Outgoing Routing information:

- Approve
- Escalate
- Reject

These show the possible outgoing routes from this step. They are configured when you set up the outgoing routes for the step (which we discuss later in the route configuration section).

Notice that, although you cannot change the outgoing routes in this step, you can specify the order in which the route conditions are evaluated. Select any one of the outgoing routes and click the up and down arrows next to it to rearrange its order in the list. Depending on how you set up the condition, sometimes the order of the condition might become important.
Clicking **Details** on the bottom of the Routing tab shows the routing condition to the next step. Figure 5-34 shows the outgoing routing conditions from the Adjustor Review step. The conditions are set when you configure individual outgoing route from the current step.

![Step Routing Conditions](image)

**Figure 5-34** Properties pane - Routing conditions

Figure 5-35 shows the main workflow map with the Adjustor Review step.

![Claim approval process main workflow map](image)

**Figure 5-35** Claim approval process main workflow map

As long as any participant selects the Approve response from the Adjustor Review step, the next step goes to the Generate Letter step. If any participant selects the Escalate response, then the next step goes to the Supervisor Review step. If the response is Reject, then the next step goes to the Claim Rejected step.

If multiple tasks can be performed in parallel after this step, you can select the option **All true conditions**.
5.4.2 Submap step

A submap step calls a submap to perform activities. Submaps can help organize complex workflows by breaking them into more manageable pieces. In addition, if part of a process is executed multiple times (with multiple steps), you can create a submap to encapsulate that part of the repeated process and have the submap steps calling the submap whenever needed in a workflow map. A main workflow map can have multiple submaps. A submap can also have other submaps.

Figure 5-35 on page 100 shows our example, the main workflow map. In the map, there is a submap step called Claim Setup. This submap step calls the Claim Setup submap as shown in Figure 5-36. The submap encapsulates all the tasks of setting up the claim, allowing the main workflow map to hide the details.

You can configure a submap step through the following tabs:

▶ General tab
▶ Attributes tab
▶ Routing tab

If you configure your workflow to use a rules engine, there is also a Rules tab. It is beyond the scope of this book to cover rules engines. We do not discuss the configuration of rules engines in this book.
General tab

Figure 5-37 shows the General tab in the Properties pane for a submap step.

![Properties pane - General tab for submap step]

The General tab allows you to configure general information about the step, such as the submap to call and the description.

From the General tab, set up the following fields:

- **Map**
  - Select the submap to be executed when workflow reaches this step. For our example in Figure 5-37, we show the properties of the Claim Setup submap step. This submap step calls Claim Setup Subprocess submap. You create the submap in similar ways as you create a main workflow map, with steps and routes.

- **Description**
  - A description of the submap.
Attributes tab

Figure 5-38 shows the Attributes tab in the Properties pane for a submap step.

![Properties pane - Attributes tab for submap step](image)

You can set up attributes to be used in the submap step for special processing. Attributes are valid only within this step. Unlike parameters, which can be used throughout the entire workflow, attributes are created and used only at the particular step.

From the Attributes tab, for each attribute that you want to use, set up the following information:

- **Name**
  - Enter a name for this attribute.

- **Type**
  - Select the attribute type, including boolean, float, integer, and string. Attribute type also supports arrays.

- **Value**
  - Enter a value for the attribute. Note that you cannot use Expression Builder for an attribute value.

Setting up attributes for a submap step is optional. For our example, we do not set up any.
Routing tab

Figure 5-39 shows the Routing tab in the Properties pane for a submap step.

In the Routing tab, you configure how the step handles the incoming routes and how it processes the outgoing routes.

From the Routing tab, set up the following fields:

- **Incoming Routing Information**
  
  If you used parallel routing earlier, at some point, you must have a collector step to combine the parallel routes into one place. If the current step is the collector step, check the **Collector Step** option. Otherwise, leave it unchecked.

- **Outgoing Routing Information**
  
  If there are multiple routes leaving from this step to other steps, you specify how the route is determined in this field. You can select one of the following options:
  
  - All true conditions: All specified conditions must be true before the route can be taken.
  
  - First true conditions: As soon as one condition becomes true, the route is taken.

  For our example as shown in Figure 5-39, we select the second option, First true conditions.
There are two options listed under the Outgoing Routing information:

- > 3K
- <= 3K

Clicking **Details** on the bottom of the Routing tab shows the routing condition that is set from this step to the next step. Figure 5-40 shows the outgoing routing conditions from the Claim Setup step. If the VehicleEstimationAmount is greater than 3000, then it goes to the Adjustor Review step. If the estimated amount is less than or equal to 3000, then it goes directly to the Generate Letter step.

![Figure 5-40 Properties pane - Routing condition for submap step](image)

**5.4.3 System step**

A system step calls system functions in a workflow definition. System functions perform specific functions. They can be grouped as follows:

- **General system functions**: Provide general system functions. See details below.

- **Checkpoint system functions**:Enable rolling back of the workflow process to a specific point in time.

- **Timer**: Provide timer constraint on parts of a workflow process that must be completed within a certain time frame. This includes setting a beginning timer, ending time, and the ability to suspend and resume the timer.

- **Web services system functions**: Provide functions to interact with Web services, including calling and receiving the Web services, and replying back to the caller.
General system functions include the following functions:

- **Assign**: Assign values to data fields.
- **DbExecute**: Executes a stored procedure in any supported database.
- **Create**: Launches a new workflow.
- **Call**: Call a submap.
- **Return**: Exit from the current submap and return to the calling map.
- **Delay**: Delay the workflow for a certain period of time.
- **Log**: Log the event in the event log.
- **TerminateBranch**: Terminate the current branch of workflow processing.
- **TerminateProcess**: Terminate the entire workflow processing.
- **WaitForCondition**: Suspend existing processing until a certain condition becomes true.

Checkpoint system functions include the following functions:

- **Checkpoint - BeginCheckpoint**
- **Checkpoint - EndCheckpoint**
- **Checkpoint - RollbackCheckpoint**

Timer system functions include the following functions:

- **Timer - BeginTimer**
- **Timer - EndAllTimer**
- **Timer - EndTimer**
- **Timer - ResumeTimer**
- **Timer - SuspendTimer**

Web services system functions include the following functions:

- **Invoke**
- **Receive**
- **Reply**

There are several tabs available where you can configure a system step:

- **General tab**
- **Attributes tab**
- **Routing tab**

The configuration of the last two tabs is similar to that of a general step and a submap step. We discuss only the general tab here.
General tab
Figure 5-41 shows the general tab in the Properties pane for a system step.

From the general tab, provide a description for the system step. For each system function that you want the system step to perform, follow these procedures:

1. Select the system function from the Available Functions list.
2. Click the right arrow icon.
3. After the system function appears in the Selected Functions list, double-click it.
4. Depending on the type of the system function you added, enter appropriate configuration information.

The best way to learn how to use various system functions is to try them out in Process Designer. The settings for the system functions are fairly intuitive. We do not cover the configuration of each of them in detail. For detailed information, refer to ecm_help.
5.4.4 Component step

Component steps enable a workflow to interact with external applications, systems, and processes. You can import custom components and use the functions they provide in your workflow.

Figure 5-42 shows the Claim Setup submap with the component steps.

There are several tabs available where you can configure a component step:

- General tab
- Attributes tab
- Routing tab

The configuration of the last two tabs are similar to that of a general step, a submap step, or a system step. We discuss only the general tab here.

**General tab**

Figure 5-43 shows the general tab in the Properties pane for a component step.
The general tab allows you to set up the component and the operations that you use at this step.

From the general tab, set up the following information:

- **Description**
  Enter a description for the component step.

- **Operations**
  Add one or more operations to be performed by components at this step.
  The order of the operations listed here is the order of operation execution. To rearrange the order, select the operation and click the up or down arrow icon on the upper right corner of the Operations entry box. To delete, click the X icon. To add, click the document icon on the upper right corner.

  The default component provided by the system is the CE_Operations. You can also add your own component operations there.

  For our example, we show the component step, Create Case. In this step, the following operations are executed:
  - `createFolder()`: Create a folder for the claim case.
  - `file()`: File all the claim related files in the claim folder.
  - `searchForOne()`: Searches for claim related files.
Notice that the first and the last operations are available through the More_Operations component. The file() operation is available through the CE_Operations component.

To add an operation, follow these procedures:

a. Click the Add (document) icon on the upper right corner of the Operation entry box. The Operation Selection dialog box appears (see Figure 5-44).

b. Select a component from the Component drop-down list.

c. Select one or more operations in the list and click OK.

![Operation Selection dialog box](image)

*Figure 5-44  Operation selection for the component step*

- **Operation Parameters**

  Enter the necessary parameter information (their names, types, and values in the expression column) required for the operations in this step. Not all operations require parameters. For our example, as shown in Figure 5-43, only the file() operation requires two parameters:

  - sourceDocument
  - destFolder

  Both of these parameters are of attachment type. The file() operation files the claim related documents (sourceDocument) to the claim folder (destFolder).

### 5.4.5 Launch step

A launch step represents the first step in a business workflow process. Configuring a launch step is similar to configuring a general step, with a few differences.
In the general tab, there is no step destination. In addition, you cannot define a deadline in a launch step. In the Assignments tab, there is only the After Completion subtab, but no Before Execution subtab. This is because the launch step is the first step in a workflow process. In the Routing tab, there is no Incoming Routing Information setting. Again, this is because the launch step is the first step and there are no incoming routes for the step.

Refer to 5.4.1, “General step” on page 91 for configuration details.

### 5.4.6 Start step

The start step is the first step on a submap. It is similar to a general step except that you cannot set a deadline for the step.

Refer to 5.4.1, “General step” on page 91 for configuration details.

### 5.5 Creating routes with Process Designer

Routes define how work moves from one step to the next in a business process.

**Adding a route**

To add a route between two steps in a workflow map, do the following procedure:

1. Move the mouse over the starting step until the route symbol appears.
2. Drag the route from the starting step to the ending step.

Figure 5-45 illustrates these steps.

**Figure 5-45** Drawing routes between steps
Configuring a route
To configure a route that you added in a workflow map:
1. Select the route in the workflow map.
2. Edit its properties in the Properties pane.

Deleting a route
To delete a route from a workflow map:
1. Right-click the route in the workflow map.
2. Select Delete from the context menu.

5.5.1 Setting conditions for routes
You can set routing conditions to either Always true or Condition.

If there is only one outgoing route from the current step to the next step, that is, if the workflow process always goes from the current step to the next step, then select Always true. Figure 5-46 shows routes in the main workflow for our example. From the Launch step to the Claim Setup step, there is only one route. Thus, for the example, we select Always true in the Properties pane. Notice that when Always true is selected, you cannot configure anything else for the route.

If there are multiple routes coming from the current step to other steps, you select Condition for the route. You can configure the routing condition based on responses that the participants select in the current step (where this route comes from) or based on values of the data fields in the workflow process. You can also use the combination of both responses and data fields to set up the routing condition.

The data fields used for conditional routes, such as data fields, attachments, and workflow groups, must be defined in the workflow properties.

With multiple outgoing routes, configure the condition for each route so the system understands under which condition a specific route is taken.
For a route that is set as Condition and not as Always true, configure it with the following tabs:

- Responses tab
- Data fields tab

**Response tab**

When a participant completes a task at a step, you can set up a list of responses that a participant can select. Each response corresponds to an outgoing route from the step.

To configure the route based on the responses, configure the following fields in the Response tab:

- **Condition**
  - **ALL**: Return true if all participants select a particular response.
  - **NONE**: Return true if none of the participants selects a particular response.
- **ANY**: Return true if any participant selects a particular response.

- **COUNT**: Return true if the total number of participants selecting a particular response is either equal to, not equal to, greater than, greater than or equal to, less than, or less than or equal to a specific value.

- **Operator**: If COUNT is selected, then it specifies the operator used to compare the count with a particular value. Operator options include equal, not equal, great than, greater than or equal, less than, and less than or equal to that value.

- **Value**: If COUNT is selected, then it specifies the value that corresponds to the number of participants who selected a particular response.

**Response**

Select the particular response that works with the specified condition above.

After setting the condition, click the **Insert** button to add the expression on the condition field. You can manually edit the condition expression also. After you insert the condition for a route on the bottom entry box, it does not matter what values are selected or shown in the Condition, Response, and other fields mentioned previously. Always make sure that the condition appears at the bottom before continuing. See Figure 5-47.
The following list shows conditional routing examples based on our Claim Approval main workflow map:

- The Approve route that connects the Adjustor Review step to the Generate Letter step has the following condition:
  
  \[ \text{ANY(Approve)} \]
  
  This means that as long as one participant selects the Approve response at the Adjustor Review step, the Approve route’s condition returns true, and the next step in the workflow goes to the Generate Letter step.

- The Escalate route that connects the Adjustor Review step to the Supervisor Review step has the following condition:
  
  \[ \text{ANY(Escalate)} \]
  
  This means that as long as one participant selects the Escalate response at the Adjustor Review step, the Escalate route’s condition returns true and the next step in the workflow goes to the Supervisor Review step.

- Using the same logic, the Reject route has the following condition:
  
  \[ \text{ANY(Reject)} \]
  
  If any one participant selects the Reject response, then the next step goes to the Claim Rejected step.

- Let us assume that we have the following condition for a route:
  
  \[ \text{COUNT(Approve)} \geq 2 \]
  
  In this case, as soon as two or more participants select the Approve response, the condition returns true and the route is taken.

Remember that you must define a list of available responses at the step from which the routes originate. At each route leading from the step, you set the condition for when each route should be taken.

**Note:** If there is only one participant processing the task at the step, then the condition ANY and ALL would be the same.

**Data Fields tab**

Instead of using responses selected by participants as the condition for a route, you can also use the values in the data fields to create a condition for evaluation.
Figure 5-48 shows the condition set for a route.

Figure 5-48  Properties pane - Conditional Routing Data Fields for a route

To set up the condition, use the following fields to construct an expression for the condition and then click **Insert**:

- **Field**: Select a data field to be used in the condition expression.
- **Operator**: Select the operator to evaluate the condition.
- **Value**: Select a value to evaluate the condition.

Remember that, after you select and set the field values, click **Insert** to add the condition for the route. You can manually edit the condition using AND, OR, and parentheses to create complex expressions.

For the example shown in Figure 5-48, we have the following condition:

EstimateVehicleAmount>3000

This means that the route is taken if the estimated repair cost for the vehicle is less than 3000.
Combining both responses and data fields

You can set the route condition to be based on both participant responses and data field values. To accomplish this, manually edit the condition expression. The following expression is an example of a condition based on both a data value and the participant's response:

\[ \text{EstimateVehicleAmount} > 3000 \text{ and ANY(Approve)} \]

**Note:** Set up condition routing carefully. If you have multiple outgoing routes from a step, yet none of the conditions in the routes returns true, the workflow will stop and produce an error.

### 5.6 Advanced topics

In this section, we discuss topics related to creating workflow definitions using Process Designer that are beyond the basic concepts we have covered so far. Specifically, we discuss workflow inheritance and how it works, default system submaps and how to override them, and using the Expression Builder to build various expressions for route conditioning or parameter assignment.

#### 5.6.1 Workflow inheritance

When workflow inheritance is enabled, you can choose an existing workflow definition as the base workflow from which a new workflow definition inherits. This allows you to leverage existing definitions at a higher level and pass them along to new workflow definitions.

A new workflow can inherit the following items from a base workflow:

- Main workflow map
- Submaps
- Data field, attachment, and workflow group definitions
- Workflow deadline and reminder
- Milestones
- Event log
- Roster
- Condition Identifier
- Partner link and XML schema
- XML data field
- Incoming Web services attachment folder
- Rule set names
- E-mail notification preference
To override the inherited items, you can redefine them in the new workflow definition.

Let us look at some examples. Figure 5-49 shows two workflows, Workflow A and Workflow M, and their local and inherited workflow properties.

**Workflow A**
- **Local:** main-A, submap-a1, submap-a2, field-a1, field-a2
- **Inherited:** main-WorkObjectEx

**Workflow M**
- **Local:** main-M, submap-a1*, submap-m1, field-m1
- **Inherited:** main-A, submap-a1, submap-a2, field-a1, field-a2

**Figure 5-49** Workflow inheritance flow hierarchy for Workflow A and M

Workflow A inherits from the default workflow definition, WorkObjectEx. Workflow A contains the following items:
- Main workflow map A
- Submaps a1 and a2
- Data fields a1 and a2

Workflow M inherits some of the items from Workflow A and overrides others. Workflow M contains the following items:
- Main workflow map M (which overrides the one from Workflow A)
- Submap a1* (which overrides the one from Workflow A)
- Submap a2 (which is inherited from Workflow A)
- Submap m1 (which is new for Workflow M)
- Data fields a1 and a2 (which is inherited from Workflow A)
- Data field m1 (which is new for Workflow M)

Now adding a new Workflow R, we have the workflow inheritance flow hierarchy for these three workflows as shown in Figure 5-50.

**Figure 5-50** Workflow inheritance flow hierarchy for Workflow A, M, and R
Workflow R inherits from Workflow M. It contains the following items:

- Main workflow map M (which is inherited from Workflow M)
- Submaps a1* and m1 (which are inherited from Workflow M)
- Submap a2 (which is inherited from Workflow A)
- Submap rm1 (which is new for Workflow R)
- Data fields a1 and a2 (which are inherited from Workflow A)
- Data field m1* (which is inherited from Workflow M)
- Data field r1 (which is new for Workflow R)

If inheritance is disabled, then Workflow R contains only the following items (see Figure 5-51):

- Main workflow map R (which is new for Workflow R)
- Submap rm1 (which is new for Workflow R)
- Data field r1 (which is new for Workflow R)

All other inherited main maps, submaps, and data fields are no longer available.

[Diagram showing Workflow A, Workflow M, and Workflow R]

Figure 5-51 Disable workflow inheritance and its affect on Workflow R

Now, let us look at an example of enabled inheritance, but switching base workflow. Again, we have Workflow A, M, and R as discussed earlier. We introduce a new Workflow N that inherits from Workflow A with a new submap and a new data field. Now, if we switch the base workflow for Workflow R from Workflow M to Workflow N — in other words, Workflow R now inherits from Workflow N — let us see what happens. Figure 5-52 shows the new resulting Workflow R.
Introducing IBM FileNet Business Process Manager

From Figure 5-52, Workflow N, which inherits from Workflow A, contains the following items:

- Main workflow map N (which is new for Workflow N)
- Submaps a1 and a2 (which are inherited from workflow A)
- Submap n1 (which is new for Workflow N)
- Data fields a1 and a2 (which are inherited from Workflow A)
- Data field n1 (which is new for Workflow N)

When inheritance is enabled and we switch the base workflow for Workflow R from Workflow M to Workflow N, Workflow R now contains the following items:

- Main workflow map N (which is inherited from Workflow N)
- Submaps a1 and a2 (which are inherited from Workflow A through N)
- Submap n1 (which is inherited from Workflow N)
- Data fields a1 and a2 (which are inherited from Workflow A through N)
- Data field n1 (which is inherited from Workflow N)
- Submap rm1 (which is new for Workflow R)
- Data field r1 (which is new for Workflow R)

By comparing the resulting main workflow map, submaps, and data fields of Workflow R now with the one earlier (which inherits from Workflow M), you can see how inheritance works.
5.6.2 Default system submaps

When defining a simple, stand alone workflow process, it might be sufficient to put all the steps on the main workflow map. However, if the workflow definition becomes more complex, it might be more practical to break up the workflow into smaller workflows (submaps) that can then be called by the main workflow map. Submaps make the workflow definition more modular. When using submaps, the main workflow map acts as a container for the submaps.

IBM FileNet Business Process Manager comes with a set of default system submaps that you can use for your workflow: Malfunction and Terminate maps. You can customize them for your business requirements.

Malfunction map
The malfunction map executes whenever an error occurs during workflow processing. When an error occurs, the malfunction map moves the work item to a system queue to be reviewed by an administrator. After the work item is reviewed and the error condition is resolved, the work item can be returned to either the calling step where the error condition occurred or it can be routed to the next step in the business process.

Figure 5-53 shows the Malfunction map in Process Designer.
In most cases, there is no need to work with the malfunction map. The system executes it only when required. By default, the Malfunction map is read-only, but you can override it to meet your business requirements. Use caution when changing the default behavior of the Malfunction map. Test thoroughly to avoid unexpected results during workflow processing.

**Terminate map**

When a process is complete, the work item is routed to the Terminate map. There are no steps associated with the Terminate map and there is no need to explicitly call it from the main workflow. Each workflow map automatically calls the Terminate map at the end of the workflow.

Figure 5-54 shows the Terminate map in Process Designer.

![Figure 5-54 Terminate map in Process Designer](image)

In most cases, there is no need to work with the Terminate map. The system executes it automatically. By default, the Terminate map is read-only, but you can override it to embed common terminating steps for your workflow process. Use caution when changing the default behavior of the Terminate map. Test thoroughly to avoid unexpected results during workflow processing.

### 5.6.3 Expression Builder

The Expression Builder dialog box is used to build expressions that can be evaluated by the workflow. The expressions are built using components that exist in the workflow definition such as data field names, attachments, workflow groups, XML data fields, system fields and partner links. In addition to workflow components, system functions can also be used in expressions. The Expression Builder dialog box can be launched in any field that permits expressions.
To build an expression, specify the following items in the Expression Builder:

- Choose a category from the category drop-down menu. This determines which items are available to be added to the expression. Categories include Data Fields, Attachments, Workflow Groups, XML Data Fields, System Fields, Partner Links, and Functions. See Figure 5-55 for the Expression Builder category drop-down menu.

![Figure 5-55 Expression Builder Category drop-down menu](image1)

- Choose the desired category items from the pick list to be added to the expression.

  Figure 5-56 shows the data fields available for the workflow.

![Figure 5-56 Expression Builder: Data Fields](image2)
Figure 5-57 shows the system fields that can be used in an expression for the workflow.

![Expression Builder: System Fields](image)

Choose appropriate operators (such as +, -, *, /, is equal, not equal, is greater than, is greater than or equal), conjunctions (AND, OR), parentheses to be added to the expression.

Click the **Insert** button to add the selected function, operator, or other selected items to the expression area at the bottom of the Expression Builder dialog.

Figure 5-58 shows the functions that are available to be used in an expression for the workflow.

![Expression Builder: Functions](image)
Modify the expression in at the bottom of the dialog as necessary.

**Note:** Click the **Clear** button to remove the content from the expression area.

You can use a combination of data fields, system fields, functions, Workflow Groups, and other categories to build the expression. If you manually edit the expression, make sure the syntax is correct.

To type literals into the expression area, enter values in the appropriate format:

- Strings must be enclosed in quotes.
- Array values must be separated by commas (,) and surrounded by braces {,}.

For more details, refer to ecm_help.
Implementing business processes: Case study

In this chapter we provide step-by-step instructions for implementing business processes with IBM FileNet Business Process Manager. We use the case study, a claim approval application of a fictitious insurance company that is introduced earlier in the book, to demonstrate how to implement the business processes.

We cover the following topics:

- Building business processes overview
- Defining the basic process object model
- Starting a new workflow process
- Building the process workflow map
- Finishing the step properties definition
- Building a submap
- Creating a workflow subscription
- Testing the basic process
- Creating and calling a Web service

**Note:** This book is written based on Version 4.0 of IBM FileNet Business Process Manager. For release 4.5 or later, some of the user interface, including menus and toolbar, might be changed.
6.1 Building business processes overview

There are many ways to build a business process using IBM FileNet Business Process Manager. We recommend following these steps:

1. Define the basic process object model.

2. Start a new workflow process:
   a. Define the workflow properties.
   b. Save the workflow process.
      Always perform this step whenever you make any major changes to ensure that your work is properly saved.

3. Build the main workflow process map:
   a. Add steps.
   b. Add routes.
   c. Define conditional routes if any.
   d. Validate the workflow process map.
      Always perform this step whenever you make any major changes to ensure that there is no error in the workflow process map.

4. Define the individual steps in the main workflow process map.
   - Define Web services steps, if any.
   - Define Component Integrator steps, if any.

5. Create and define any submap used in the main workflow process.

6. Create subscriptions to launch the workflow process.
   This is an optional step. You can invoke your workflow process in multiple ways. We show you one way of creating a workflow subscription in a later section of the chapter.

Not all the steps have to be created in the sequence we describe here. For example, it is common to create a simple workflow map, with all the steps and routes first drawn in Process Designer, before going into more detail of defining each step. This is a top-down approach where you define all the elements (steps and routes) on the map graphically and then define each element in detail. Another approach is to define each element in detail as you build the entire map. In addition, you can decide when to build the submap and when to build the Web services or Component Integrator steps. Based on your preferences, you decide how you want to build your workflow.

In the following sections, we describe the detailed steps required to build business processes.
6.2 Defining the basic process object model

The claims approval process integrates content (claims and supporting documents) with a business process. Content is permanently stored as objects in the IBM FileNet Content Engine. The workflow process runs in the Process Engine and has its own temporary storage in the Process Engine while the process is active. To permanently store and update information, the workflow process can use and update the data persistently stored in the Content Engine.

To store content in Content Engine, you must first define the object model using the IBM FileNet Enterprise Manager. It is beyond the scope of the book to show you how to define the object model step-by-step. However, for your convenience, we provide the XML data files that you can import into the Content Engine so that you can follow the steps we show here for our case study.

For our case study, we have to define the following components:

- Document object model for supporting documents
- Document entry template for supporting documents

Document object model for supporting documents

The claim approval process requires a document object to be created in the IBM FileNet P8 Content Engine. Create the document object from the IBM FileNet Enterprise Manager tool using the configuration outlined in Table 6-1 through Table 6-5:

- Table 6-1 and Table 6-2 show the Supporting Claim Document class definition, including the class properties information.
- Table 6-3 and Table 6-4 show the choice list for different supported document types.
- Table 6-5 shows the storage policy information.

Table 6-1  Supporting Claim Document class object

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbolic Name</th>
<th>Description</th>
</tr>
</thead>
</table>

Table 6-2  Supporting Claim Documents class properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbolic name</th>
<th>Description</th>
<th>Choice list</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClaimID</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Last Name</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>DocType</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
The claims approval process requires the creation of a Document Entry Template for supporting documents. Using the Advanced Tools in the IBM FileNet Workplace, create a Document Entry Template with the following parameters. For more information on creating Document Entry Templates, refer to the IBM FileNet Workplace Administration Guide.

Define an entry template. Set the folder, properties and security as shown in Table 6-6.

Table 6-6 Document Entry Template Folder properties

<table>
<thead>
<tr>
<th>Object Store</th>
<th>P8Demo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folder</td>
<td>Incoming</td>
</tr>
<tr>
<td>Hide Select Folder step</td>
<td>selected</td>
</tr>
<tr>
<td>Constrain to folder selected or it sub-folders</td>
<td>unchecked</td>
</tr>
</tbody>
</table>
Modify the Supporting Claim Document Class with the properties shown in Table 6-7.

Table 6-7  Supporting Claim Document Class

<table>
<thead>
<tr>
<th>Class</th>
<th>Supporting Claim Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Set Properties step</td>
<td>selected</td>
</tr>
<tr>
<td>Yes, auto classify the content on entry and use base Document Class by default</td>
<td>unchecked</td>
</tr>
<tr>
<td>Yes, use entry template for check-in</td>
<td>checked</td>
</tr>
</tbody>
</table>

Set the Document Entry Template properties as shown in Table 6-8. Set the Entry Template security settings as shown in Table 6-9.

Table 6-8  Document Entry Template properties

<table>
<thead>
<tr>
<th>Required</th>
<th>Property</th>
<th>Default Value</th>
<th>Access Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>unchecked</td>
<td>Document Title</td>
<td>{FILENAME}</td>
<td>Hide</td>
</tr>
<tr>
<td>unchecked</td>
<td>ClaimID</td>
<td></td>
<td>Editable</td>
</tr>
<tr>
<td>unchecked</td>
<td>Last Name</td>
<td></td>
<td>Editable</td>
</tr>
<tr>
<td>unchecked</td>
<td>DocType</td>
<td></td>
<td>Editable</td>
</tr>
<tr>
<td></td>
<td>Add as major version:</td>
<td>Yes</td>
<td>Hide</td>
</tr>
</tbody>
</table>

Table 6-9  Document Entry Template Security

<table>
<thead>
<tr>
<th>Title</th>
<th>Owner Control</th>
<th>Promote Version</th>
<th>Modify Content</th>
<th>Modify Props</th>
<th>View Content</th>
<th>View Props</th>
<th>Publish</th>
<th>Remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>#AUTHENTICATED-TEOUSERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#CREATOR-OWNER</td>
<td>checked</td>
<td>checked</td>
<td>checked</td>
<td>checked</td>
<td>checked</td>
<td>checked</td>
<td></td>
<td>unchecked</td>
</tr>
<tr>
<td>Administrator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Set the Security step display option to: Hide Set Security step. See Table 6-10.

**Table 6-10  Set security step**

<table>
<thead>
<tr>
<th>Hide Set Security step</th>
<th>selected</th>
</tr>
</thead>
</table>

Under option 6, select Folder and Document Title. See Table 6-11.

**Table 6-11  Setting a folder for the Document Entry Template**

<table>
<thead>
<tr>
<th>Setting under option 6</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Folder</td>
<td>***BPM Lab → Template</td>
</tr>
<tr>
<td>Document Title</td>
<td>Claim Supporting Document Entry Template</td>
</tr>
</tbody>
</table>

**6.3 Starting a new workflow process**

To create a workflow process, use the Process Designer application. Follow these steps to launch Process Designer and start a new workflow process:

1. Launch Internet Explorer and enter the following URL:
   
   http://<host name>:<port>/Workplace

   Where:
   
   –  `<host name>` is the name of the machine where Application Engine is installed.
   –  `<port>` is the port number where the Workplace application resides.

   For our case study, we enter:
   
   http://queens:9080/Workplace

   The Workplace logon window appears.

2. Log in using an appropriate user ID and password.
3. From the left-hand pane, select **Author → Advanced Tools → Process Designer** as shown in Figure 6-1.

![Image of FileNet Workplace with Process Designer highlighted](image-url)

**Figure 6-1  Launching Process Designer**
Introducing IBM FileNet Business Process Manager

This launches Process Designer and automatically starts a new workflow process. See Figure 6-2.

![Figure 6-2 Process Designer: start a new workflow process](image)

6.3.1 Defining workflow properties

Workflow properties contain global variables and property values that are applicable to the entire workflow process. You can define them in the beginning of the workflow creation process. Alternatively, you can modify or add information later.

To define workflow properties, follow these steps:

1. From Process Designer, select **View → Workflow Properties**. See Figure 6-3.

   **Tip:** Roll your mouse over the tool bar below the main menu to locate the shortcuts to frequently used functions, such as the Workflow Properties icon.
The Workflow Properties dialog box opens as shown in Figure 6-4.

2. From the Workflow Properties dialog box, on the **General** tab:
   
   - Enter the Workflow Name, Subject, and Description.
     
     For our case study, we enter **XYZ Basic Process** for the Workflow Name. We leave the other fields blank.
   
   - Enter Deadline information. You can specify within how long (in minutes, for example) the workflow must complete and within what time frame (in minutes, for example) before the deadline comes to send a reminder.
     
     For our case study, we do not use the Deadline feature. We leave these fields blank.
3. From the Workflow Properties dialog box, click the **Advanced** tab. The panel shown in Figure 6-5 opens. In this tab:

   - Make selections for Roster, Event Log, and Condition Identifier.
     
     For our case study, we keep the default values as-is.
   
   - Select to enable or disable e-mail notification.
     
     Enable this feature if you want to notify users via e-mails. Whether you use it or not depends on your business requirements.
     
     For our case study, we do not use e-mail notification. Clear the box.

![Figure 6-5  Workflow Properties: Advanced tab](image)

4. From the Workflow Properties dialog box, click the **Data Fields** tab. This is where you specify all the data fields (parameters) that are used for the workflow process. In this tab:

   - Add the data fields used for your workflow process. For each data field, double-click the Name cell to enter its value and press Enter. Use the drop-down box under the Type to select the data type. Also, enter information for Merge Type, Expression, and Description. If you decide later that you need additional fields, you can come back to this tab and add more fields.

     For our case study, we enter the data fields as shown in Figure 6-6.

     **Tip:** Before adding a number of data fields of the same type, change the data field type to the one you use the most. Each subsequent data field entry automatically uses this type. For our case study, we use String frequently, so we change the Type to String, and then add the rest of the data fields.

     There is always one extra line in the end. Ignore that line.
5. From the Workflow Properties dialog box, click the **Attachments** tab. This is where you specify all the attachments that can be added during the workflow process. Attachments can include folders, IBM FileNet P8 documents and objects, and external documents that are a part of the work item for the workflow process. In this tab:

- Add attachments used for the workflow process. For each attachment, enter Name, Value, and Description.

  For our case study, we enter attachment information as shown in Figure 6-7.

- If there are multiple attachments of the type specified, select the **Array** check box. If there is only one attachment of the specified type, leave the check box blank.

  For our case study, we might have multiple SupportingDocs. We select the **Array** check box for that attachment type. For other types of the attachments, such as ClaimForm and PolicyDoc, we leave that check box blank.
– If a particular attachment initiates the workflow process, select the attachment entry and click the blue **Initiating Attachment** icon on the toolbar. This results an arrow appearing next to the entry.

For our case study, the creation of a ClaimForm initiates the workflow. We therefore set it up as the **Initiation Attachment**. See Figure 6-7.

![Figure 6-7 Workflow Properties: Attachments tab](image)

– To enter a value for the attachment, double-click the value cell and select an item from an existing object store.

For our case study, the workflow process uses a search template to search for claims during one of its tasks. We need to add this search template as an attachment for the workflow. We call this attachment ClaimSearchTemplate and it uses a search template that is already created in the system as its value. The existing search template, called Search for Claim ID, searches from the system and returns the claims with a given Claim ID. To specify the existing search template (Search for Claim ID) as the value for the attachment (ClaimSearchTemplate), use the following steps:

i. Select the attachment, for our case study, **ClaimSearchTemplate**.

ii. Double-click the corresponding Value entry. This displays a Browse for item window as shown in Figure 6-8.
iii. Navigate to the location where the search template (Search for Claim ID) is stored. In our scenario, double-click **objectstore** → **Case Study Searches**, and select **Search for Claim ID**. See Figure 6-9.

**Note:** Our case study is built in an object store called objectstore. The existing search template is in the Case Study Searches folder. You can create your search template in any folder within your object store and navigate to that location. If you import the object store data provided by this book, use this folder.

iv. Click **Select**.

**Note:** This attaches a reference to the Search Template process parameter.

For our case study, we also select values for other attachments as shown in Table 6-12.
Table 6-12  Attachments and values

<table>
<thead>
<tr>
<th>Name</th>
<th>Browse path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParentFolder</td>
<td>objectstore → Claims</td>
<td>This is the root folder where all claims will be stored.</td>
</tr>
<tr>
<td>SearchSupportingDocs</td>
<td>objectstore → Case Study Searches → SupportingDoc via LastName</td>
<td>This is a search template that searches all supporting documents via a last name.</td>
</tr>
<tr>
<td>SearchPolicy</td>
<td>objectstore → Case Study Searches → Search for Policy</td>
<td>This is a search template that searches for related policies.</td>
</tr>
<tr>
<td>WordTemplate</td>
<td>objectstore → Case Study Templates → Claims Processed Letter Template</td>
<td>This is a Microsoft Word template that is used to create a standard Claims Processed letter.</td>
</tr>
</tbody>
</table>

6. From the Workflow Properties dialog box, click the **Workflow Groups** tab. This is where you specify the workflow groups that are used in a workflow process. A workflow process has multiple steps. Each step can have one or more participants performing tasks. The participant can be an individual or a group of people. If it is a group of people, you define the group of people as the workflow group in this tab.

Enter the workflow group name by selecting the Name cell and typing in the workflow group name or use the pencil icon to add or modify the workflow group:

a. Click the **Modify** (pencil) icon. See Figure 6-10.

Figure 6-10  Workflow Groups tab
b. Select **Users** or **Groups**. Type the first few characters for the user name or the group name in the Starts with field and click the **Search** (magnifying glass) icon. See Figure 6-11, where we typed a in the Starts with field.

![Figure 6-11: Search for a user or group](image)

**Figure 6-11**  Search for a user or group

c. A list of users or groups appears that matches the characters you typed in the Starts with field. Select the users or workflow groups and click the **Add** button. See Figure 6-12.

![Figure 6-12: Add a user or a group to the workflow participant list](image)

**Figure 6-12**  Add a user or a group to the workflow participant list

d. Click **OK**.

For our case study, we add Adjustor, Field_Agent, and CreatedBy as the workflow groups as shown in Figure 6-13.

![Figure 6-13: Participant Selection](image)
From the Workflow Properties dialog box, there are other tabs including the Maps, Milestones, and Web services tabs. In these tabs, you can specify other workflow maps that you use, any milestones you want to establish, and any Web services that you want to use within the workflow process.

For simplicity of the first basic workflow, we do not define anything in these tabs at this point in time for our case study.

When everything is defined in the Workflow Properties dialog box, click Close.

6.3.2 Saving the workflow process

Now is a good time to save what you have created and defined so far. There are three methods of saving a workflow process:

- Option 1: Save the workflow process definition locally on your workstation or a share drive using File → Save As.
- Option 2: Check in the workflow process definition into Content Engine and check out again for update using File → IBM FileNet Add New.
- Option 3: Hybrid of the above two methods using File → IBM FileNet Save.

Option 1: File -> Save As

The first option, saving your workflow process locally, allows you to develop and test the process to completion before checking it into Content Engine. To save the workflow process locally, from Process Designer, select File → Save As or File → Save.
Option 2: IBM FileNet Add New
The second option is to check the workflow process into Content Engine and create a new version when the workflow process is checked in. This allows Content Engine to keep track of the changes with a version number. To check in the workflow process for the first time, follow these steps:

1. From Process designer, select **File → IBM FileNet Add New**. See Figure 6-14.

   **Note:** Use **IBM FileNet Add New** for the first time check-in only. After the initial check-in, use **IBM FileNet Open/Checkout**, **IBM FileNet Checkin**, and **IBM FileNet Cancel Checkout** functions to manage the versioning of the workflow process.

2. The Save the workflow definition to an object store dialog box displays as shown in Figure 6-15. Click **Browse** and navigate to a folder where you want to store the workflow process. To navigate, select a folder and click **Open** or double-click the folder.
For our case study, we browse to the **Case Study Process** folder as shown in Figure 6-16 and click **Select**.

3. After the folder is selected (see Figure 6-17), click **Next**.

4. The dialog box prompts you to enter a document title for the workflow process. See Figure 6-18. Enter a descriptive name and click **Finish**.
Figure 6-18 Saving workflow process: Finish

**Option 3: File → IBM FileNet Save**

A third option is somewhat of a hybrid of the first two. Using File → IBM FileNet Save, you save the changes to Content Engine without checking in the process. When you are ready to check in as a version, you can then use option 2 to check in the workflow process.

**Tip:** To test your workflow process, you can click the **Launch** icon on the toolbar (the icon on the far right).

To keep Process Designer from adding or checking in a process (as a new version) whenever you launch the workflow process, change your preference as follows:

1. From Process Designer, select **Tools → Preferences**. See Figure 6-19.
2. Click the **Workflow** tab.
3. Clear the **Add/Check In Workflow Before Transfer/Launch** option.

Notice that from the preference dialog shown in the figure, you can also set whether you want the system to automatically validate your workflow before you transfer or launch the workflow.

Figure 6-19 Process Designer: Preference setting
6.4 Building the process workflow map

A workflow process consists of multiple steps and routes. In this section, we show you how to build a process workflow map.

6.4.1 Adding steps

At the bottom left of the Process Designer window, the step icons are arranged in palettes based on their functions. Figure 6-20 shows some of the icons.

The default palette is the BPM Palette. To see steps that are available for a different palette, click the drop-down arrow next to the palette name and select a different palette. Provided palettes include CheckPoint Palette, General System Palette, Timer Palette, and Web services Palette. You can also create your own palette for convenience of workflow design.
To add steps to a map, follow these procedures:

1. Add a new step to a map using one of the following methods:
   - Select from the pop-up menu.
     Place your cursor anywhere on the map. Right-click to display the pop-up menu. Select one of the available steps from the pop-up menu. See Figure 6-21.
   - Use drag-and-drop.
     From the palette on the bottom of the window, drag and drop the step to the map above.

**Note:** The pop-up menu from the first method displays only a list of steps that are available from the BPM palette. To add steps other than what are available from the BPM palette, use the drag-and-drop method.

For our case study, we right-click at the map area and select New Submap Step from the pop-up menu.
2. Enter a name for the step in the Properties pane’s Name field located at the right side of the Process Designer menu. See Figure 6-22.

Each step is given a default name. You can modify this name at a later time. For our case study, we type **Claim Setup**.

![Define step name](image)

**Figure 6-22   Define step name**

3. Add the rest of the steps for your map.

For our case study, we add six steps as listed in Table 6-13. We arrange the steps as shown in Figure 6-23.

*Table 6-13   Steps created for our case study*

<table>
<thead>
<tr>
<th>Step name</th>
<th>Step type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim Setup</td>
<td>Submap</td>
</tr>
<tr>
<td>Adjustor Review</td>
<td>General</td>
</tr>
<tr>
<td>Supervisor Review</td>
<td>General</td>
</tr>
<tr>
<td>Claim Processed</td>
<td>General</td>
</tr>
<tr>
<td>Claim Rejected</td>
<td>General</td>
</tr>
<tr>
<td>Generate Letter</td>
<td>Component</td>
</tr>
</tbody>
</table>
Deleting steps
If you want to delete a step from the map, right-click the step in the map and select Delete from the pop-up menu.

6.4.2 Adding routes

After adding steps to the map, you can draw routes to connect the steps together. This can be done while you add steps to the map, or you can do it after you add all the steps to the map.

To draw a route from one step (the from step) to another step (the target step):
1. Place the mouse cursor at the edge of the from step so that the cursor changes to the one shown in Figure 6-24.
2. Left-click, hold, and drag the cursor to the target step and release the mouse button.
3. Enter a name for the route in the Properties pane Name field located at the right side of the Process Designer. The route name becomes the label of the route on the map. Naming a route is optional.

   **Tip:** If there are multiple routes leading from one step to other steps, we recommend providing a descriptive name for each route. This makes the map easier to read and understand.

4. Add routes between all the steps as designed for your workflow process.

For our case study, we create routes as listed in Table 6-14. Figure 6-24 shows the setup of our case study at this point.
### Table 6-14 Routes created for the case study

<table>
<thead>
<tr>
<th>From Step</th>
<th>Target Step</th>
<th>Route name (Label)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LaunchStep</td>
<td>Claim Setup</td>
<td></td>
</tr>
<tr>
<td>Claim Setup</td>
<td>Adjustor Review</td>
<td>&gt; 3K</td>
</tr>
<tr>
<td>Claim Setup</td>
<td>Generate Letter</td>
<td>&lt;= 3K</td>
</tr>
<tr>
<td>Adjustor Review</td>
<td>Generate Letter</td>
<td>Approve</td>
</tr>
<tr>
<td>Adjustor Review</td>
<td>Supervisor Review</td>
<td>Escalate</td>
</tr>
<tr>
<td>Adjustor Review</td>
<td>Claim Rejected</td>
<td>Reject</td>
</tr>
<tr>
<td>Supervisor review</td>
<td>Generate Letter</td>
<td>Approve</td>
</tr>
<tr>
<td>Supervisor review</td>
<td>Claim Rejected</td>
<td>Reject</td>
</tr>
<tr>
<td>Generate Letter</td>
<td>Claim Processed</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 6-25 Completed routes**
Notice that if a step has multiple routes, the route is automatically numbered based on the order in which you draw the route on the map.

**Deleting routes**
If you want to delete a route from the map, right-click the route and select **Delete** from the pop-up menu.

### 6.4.3 Setting up conditional routing

A step can have multiple routes (outgoing routes) connect from it to other steps. An example of such a step is the Adjustor Review step from our case study, where there are three outgoing routes, Approve, Escalate, and Reject.

If a step has multiple outgoing routes, you can specify the conditions with which the routings happen, as follows:

1. Define the step's routing responses if user responses are involved.
2. Configure the conditional routing to be based either on user responses or on data fields.

**Defining a step’s routing responses**

If a user decides what to do with a work item at a particular step, then the user's decision at this step controls where the work item is going to be routed to. This means that you have to define a list of acceptable user responses at this step. These responses have a one-to-one correspondence with the outgoing routes from this step to other steps in the process.

To define the step’s responses used for routing:

1. Select the step.
   - For our case study, select the **Adjustor Review** step.
2. In the Properties pane on the right side, click the **Routing** tab.
3. Double-click the first cell under the Name column in the shaded Responses section and enter your response. Then press Enter.
   - For our case study, enter **Approve** in the Name column.
4. Continue to enter other responses, one for each route that connects from this step to other step.

**Note:** Remember, when entering responses in the Name column, to press Enter after each entry to ensure that the value is registered.
For our case study, we configure the user responses for two steps as shown in Table 6-15. Figure 6-26 shows the setup of our case study at this point.

Table 6-15  Configure step responses

<table>
<thead>
<tr>
<th>Step</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjustor Review</td>
<td>Approve, Escalate, Reject</td>
</tr>
<tr>
<td>Supervisor Review</td>
<td>Approve, Reject</td>
</tr>
</tbody>
</table>

Figure 6-26  Configure step responses
Configuring conditional routing

If a step has only one outgoing route, then when a person or a system completes the step, the work continues to the next step automatically. If a step has multiple outgoing routes, we have to place conditions on the routes to allow the work from this step to pass onto the correct step.

Routing of work can be based on data field values used in the process or based on user responses.

Conditional routing based on user responses

As mentioned earlier, the system can route the work item to the appropriate step based on the user’s response. For example, at the Adjustor Review step in our case study, if a user decides to accept the claim (which means that the user selects Accept at the step), the work is routed to the Generate Letter step. If the user decides to escalate the claim, then the work is routed to the Supervisor Review step. If the user decides to reject the claim, then the work is routed to the Claim Rejected step.

To configure conditional routing based on a user’s response:

1. Select the route you want to configure.
   For our case study, click the route between the Adjustor Review step and the Generate Letter step.
2. In the Properties pane, click the Condition radio button.
3. Select the Responses tab and set up the following conditions:
   a. From the Condition drop-down list, select one of these conditions: ALL, NONE, ANY, COUNT.
      For our case study, select ANY.
   b. From the Response drop-down list, select a response you defined earlier.
      For our case study, select Approve.
   c. Click Insert.
      If you make a mistake, you can click Clear to clear the routing condition and restart again. Always remember to click Insert or else the condition is not registered in the system.

Note: The drop-down list should provide a list of the responses you defined earlier for the step where this route comes from. If a response you expected is missing from the list, probably you did not press Enter earlier to register the response. Go back to the step and add the missing response.
Figure 6-27 shows the setup of our case study at this point.

Figure 6-27  Configure conditional routing based on user responses

For our case study, we set up conditional routings based on user responses as listed in Table 6-16.

Table 6-16  Configure conditional routing based on user responses

<table>
<thead>
<tr>
<th>From step</th>
<th>Target step</th>
<th>Condition</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjustor Review</td>
<td>Generate Letter</td>
<td>ANY</td>
<td>Approve</td>
</tr>
<tr>
<td>Adjustor Review</td>
<td>Supervisor review</td>
<td>ANY</td>
<td>Escalate</td>
</tr>
<tr>
<td>Adjustor Review</td>
<td>Claim Rejected</td>
<td>ANY</td>
<td>Reject</td>
</tr>
<tr>
<td>Supervisor review</td>
<td>Generate Letter</td>
<td>ANY</td>
<td>Approve</td>
</tr>
<tr>
<td>Supervisor review</td>
<td>Claim Rejected</td>
<td>ANY</td>
<td>Reject</td>
</tr>
</tbody>
</table>
Conditional routing based on data field values

In addition to routing a work item based on a user’s responses, you can also set up the routing based on data field values. For example, at the Claim Setup step in our case study, if the estimated claim amount (VehicleEstimateAmount) is more than $3000, the system routes the work to the Adjustor Review step, which enables an adjustor to manually approve, reject, or escalate the claim to a different step based on the adjustor’s review. Otherwise, if the estimated claim amount is less than or equal to $3000, the system automatically routes the claim to the Generate Letter step.

To configure conditional routing based on data field values:

1. Click the route you want to configure.

   For our case study, click the route between the Claim Setup step and the Adjustor Review step.

2. In the Properties pane, click the **Condition** radio button.

3. Click the **Data Fields** tab and set up the following conditions:
   a. From the Field drop-down list, select the data field you use to route the work to other steps.
   b. From the Operator drop-down list, select the operator you want to use.
   c. Enter a value in the Value box.
   d. Click **Insert**.

   For our case study, we want the conditional routing to happen when VehicleEstimateAmount > 3000. We set up our case study as follows:
   - Select **VehicleEstimateAmount** from the Field drop-down list.
   - Select **is greater than** from the Operator drop-down list.
   - Enter 3000 for the Value box.

   **Note:** The Field drop-down list shows a list of the data fields you set up earlier in the Workflow Properties dialog box. If a data field you require is not on the list, go back to the Workflow Properties and add the data field there.

For our case study, we set up conditional routing based on data field values as listed in Table 6-17. Figure 6-28 shows the setup of our case study at this point.
Table 6-17  Configure conditional routing based on data field values

<table>
<thead>
<tr>
<th>From Step</th>
<th>Target Step</th>
<th>Field</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim Setup</td>
<td>Adjustor Review</td>
<td>VehicleEstimateAmount</td>
<td>Is greater than</td>
<td>3000</td>
</tr>
<tr>
<td>Claim Setup</td>
<td>Generate Letter</td>
<td>VehicleEstimateAmount</td>
<td>Is less than or equal</td>
<td>3000</td>
</tr>
</tbody>
</table>

6.4.4 Validating the workflow process map

Validating your map allows you to catch any errors in the workflow process map.

**Note:** We also recommend validating your map after you make any significant change to the map.
There are two methods to validate a workflow process map from Process Designer:

- Use the menu selection.
- Use the toolbar icon.

To validate a workflow process map using the menu selection, select **File → Validate** as shown in Figure 6-29.

![Figure 6-29 Workflow validation via menu](image1.png)

Or, you can use the second method to validate a workflow process map, by clicking the **Validate** icon on the toolbar, as shown in Figure 6-30.

![Figure 6-30 Workflow validation via toolbar icon](image2.png)
If you built your workflow process map according to our case study, you should receive three errors. One error is caused because you have not defined the submap yet. The other errors are caused because you have not yet added the participants to the Adjustor Review and Supervisor Review steps. Figure 6-31 shows a sample error dialog box.

![Image of Validation Results dialog box]

Figure 6-31 Validation error

If you receive more than the three errors mentioned, use the information in the error dialog box to correct the problem before you continue with the rest of the chapter.

### 6.5 Finishing the step properties definition

After building the workflow process map with steps and routes, you get an image of what the process looks like visually. Now, you need to complete the definitions of each step involved.

**Note:** You can define the properties of steps at the same time when you build the workflow process map. From our experience, it is best to create the map first and go back to define the steps at a separate time. The sequence of building the complete workflow process depends on your preference.

You can define the properties of each step at the Properties pane on the right side of the Process Designer window.
Depending on the type of a step, you define different information. For example:

- For the General step, there are five tabs where you can enter information:
  - **Parameters**: Specify parameters to be used in the step and the operation for the step if any.
  - **Assignments**: Specify data field assignments, attributes, and milestone before and after the step execution.
  - **Routing**: Specify routing responses, and incoming and outgoing routing information.
  - **General**: Specify the instruction provided to users at the step, the step destination (either participants or working queue), participant privileges, and step processor.
  - **Deadline**: Specify the deadline information.

- For the Submap step, there are three tabs:
  - **General**: Specify the associated submap name and description.
  - **Attributes**: Specify the attributes used for the submap.
  - **Routing**: Specify the incoming and outgoing routing information.

To define each step:

1. Select the step from the workflow process map.
2. From the Properties pane on the right side, go through each tab.
3. Enter or select the necessary information.

For our case study, we set up properties for the following steps:

- LaunchStep
- Adjustor Review
- Supervisor Review
- Claim Rejected
- Claim Processed
- Generate Letter - A component step
Setting up LaunchStep
To set up Launch step in our case study:

1. Select LaunchStep from the workflow process map.

2. From the Properties pane on the right side, select the Parameters tab. Select the parameters that this workflow process uses from the Available Parameters list and click Add. The parameters should show up on the Selected Parameters on the right side. You can select multiple parameters by holding down the Ctrl button and clicking each parameter or use the Shift button.

   For our case study, we select the parameters for LaunchStep as shown in Table 6-18. Figure 6-32 shows the setup of our case study visually.

   Table 6-18 Parameters for LaunchStep

<table>
<thead>
<tr>
<th>Available parameters</th>
<th>Available parameters</th>
<th>Available parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccidentDate</td>
<td>DebitAccountNumber</td>
<td>LastName</td>
</tr>
<tr>
<td>AccidentDescription</td>
<td>EffectiveDate</td>
<td>LossLocation</td>
</tr>
<tr>
<td>Agency</td>
<td>Email</td>
<td>PolicyNumber</td>
</tr>
<tr>
<td>AgencyPhone</td>
<td>ExpirationDate</td>
<td>ReportedDate</td>
</tr>
<tr>
<td>AgentName</td>
<td>FirstName</td>
<td>VehicleEstimateAmount</td>
</tr>
<tr>
<td>ClaimForm</td>
<td>Injury</td>
<td>VehicleReplacement</td>
</tr>
<tr>
<td>ClaimSettlementType</td>
<td>InsuredNameAddress</td>
<td>VehicleVIN</td>
</tr>
<tr>
<td>Company</td>
<td>InsuredSSN</td>
<td></td>
</tr>
</tbody>
</table>
3. Select the **Assignments** tab. This is where you set the milestone for this step and assign fields and attributes with values and expressions.

   For our case study, after the completion of LaunchStep, we want to set the field **F_Subject** to be "New Claim : Claim ID "+ClaimID with quotes. To set this, we follow this procedure:

   a. Under the shaded Field Assignments section, double-click the empty cell under Name and enter the field name, **F_Subject**.
b. Under Expression, click the input box and then double-click the **Expression Builder** button on the right of the input box (see Figure 6-33).

![Figure 6-33  F_Subject assignment](image)

Figure 6-33  F_Subject assignment

C. The Expression Builder window opens as shown in Figure 6-34. Enter the following expression with quotes on the bottom pane and click **OK**:

"New Claim: Claim ID "+ClaimID

![Figure 6-34  Building expression for F_Subject field](image)
d. Enter the remaining three expressions using the data from Table 6-19.

**Table 6-19  LaunchStep - After completion field assignments**

<table>
<thead>
<tr>
<th>Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_Subject</td>
<td>&quot;New Claim: Claim ID &quot;+ClaimID</td>
</tr>
<tr>
<td>LossCity</td>
<td>substr(LossLocation, 1, strloc(LossLocation,&quot;,&quot; ) - 1)</td>
</tr>
<tr>
<td>LossState</td>
<td>substr(LossLocation, strloc(LossLocation,&quot;,&quot; ) + 1)</td>
</tr>
<tr>
<td>CreatedBy</td>
<td>{Creator}</td>
</tr>
</tbody>
</table>

**Setting up the Adjustor Review step**

To set up the Adjustor Review step in our case study:

1. Click the **Adjustor Review** step on the main map.
2. From the Properties pane on the right side, click the **General** tab. In this tab, you can enter the following information:

   - **Instruction**: Provide instructions in the user interface to participants of the step to guide them and help them in completing the step.

   - **Step Destination**: Specify at which destination the task in the step can be performed. Step destination includes Work Queue or participants. Participants can include users, user groups, or workflow groups defined in the workflow properties.

   - **Participant Privileges**: Specify the privileges the participants have when performing this step. The available option includes Reassign, View Status, and View History.

   - **Step Processor**: Specify the type of the user interface to use when participants performing this step.
For our case study:

- Under Instruction, enter the following text as shown in Figure 6-35:
  
  Please review the attached claim and related documents and approve or reject the claim.

![Figure 6-35 Adjustor Review - General tab, Instructions](image)

- This step can be accessed by users under the CreatedBy workflow group. To set this up, under the Step Destination:
  i. Select **Participants** and click the **Modify** (pencil) icon. The participant selection dialog box opens (see Figure 6-36).
  
  ii. Select **Workflow Groups**.
  
  iii. Select CreatedBy from the Available Participants, click **Add** to add it to the Selected Participants list.
  
  iv. Click **OK**.

**Note:** In reality, only participants belong to the Adjustor workflow group should be able to work on this step. For simplicity of the case study, we select CreatedBy here.
e. The participants in this step do not have rights to reassign the task, or to view the workflow status or history. To set this up, under Participant Privileges, clear Reassign, View Status, and View History.

3. From the Properties pane on the right side, click the Parameter tab. For our case study, we select the parameters for the Adjustor Review step as listed in Table 6-20.

**Table 6-20 Adjustor review parameters**

<table>
<thead>
<tr>
<th>Available parameters</th>
<th>Available parameters</th>
<th>Available parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccidentDescription</td>
<td>EffectiveDate</td>
<td>LossCity</td>
</tr>
<tr>
<td>Agency</td>
<td>Email</td>
<td>PolicyNumber</td>
</tr>
<tr>
<td>AgencyPhone</td>
<td>ExpirationDate</td>
<td>ReportedDate</td>
</tr>
<tr>
<td>AgentName</td>
<td>FirstName</td>
<td>VehicleEstimateAmount</td>
</tr>
<tr>
<td>CaseFolder</td>
<td>Injury</td>
<td>VehicleReplacement</td>
</tr>
<tr>
<td>ClaimSettlementType</td>
<td>InsuredNameAddress</td>
<td>VehicleVIN</td>
</tr>
<tr>
<td>Company</td>
<td>InsuredSSN</td>
<td></td>
</tr>
<tr>
<td>DebitAccountNumber</td>
<td>LastName</td>
<td></td>
</tr>
</tbody>
</table>
Setting up the Supervisor Review step

For our case study, we set up the Supervisor Review step as follows:

- The General tab:
  - Instruction: Please review the attached claim.
  - Participants: Workflow Groups, CreatedBy
  - Participant Privileges: Clear all

- Parameters (see Table 6-21).

Table 6-21 Parameters for the Supervisor Review step

<table>
<thead>
<tr>
<th>Available parameters</th>
<th>Available parameters</th>
<th>Available parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgencyPhone</td>
<td>EffectiveDate</td>
<td>LossLocation</td>
</tr>
<tr>
<td>AgentName</td>
<td>Email</td>
<td>PolicyDoc</td>
</tr>
<tr>
<td>CaseFolder</td>
<td>ExpirationDate</td>
<td>ReportedDate</td>
</tr>
<tr>
<td>ClaimID</td>
<td>FirstName</td>
<td>VehicleEstimatAmount</td>
</tr>
<tr>
<td>ClaimSearchTemplate</td>
<td>Injury</td>
<td>VehicleReplacement</td>
</tr>
<tr>
<td>ClaimSettlementType</td>
<td>InsuredNameAddress</td>
<td>VehicleVIN</td>
</tr>
<tr>
<td>Company</td>
<td>InsuredSSN</td>
<td></td>
</tr>
<tr>
<td>DebitAccountNumber</td>
<td>LastName</td>
<td></td>
</tr>
</tbody>
</table>

Note: When the claim is filed, the loss location information, which includes the loss city and state, is entered. The system parsed the location information from the Loss Location field to the Loss City and Loss State fields. At this step, we only need the Loss City information to process. Therefore, the parameters required at this step include only Loss City and not Loss Location or Loss State.
Setting up the Claim Rejected step
For our case study, we set up the Claim Rejected step as follows:

- The General tab:
  - Instruction: Claim has been rejected.
  - Participants: Workflow Groups, CreatedBy
  - Participant Privileges: Clear all

- Parameters:
  - Agency
  - AgencyPhone
  - EffectiveDate
  - ExpirationDate
  - LastName
  - PolicyNumber

Setting up the Claim Processed step
For our case study, we set up the Claim Processed step as follows:

- The General tab:
  - Instruction: Claim has been processed.
  - Participants: Workflow Groups, CreatedBy
  - Participant Privileges: Clear all

- Parameters (see Table 6-22).

Table 6-22  Parameters for the Claim Processed step

<table>
<thead>
<tr>
<th>Available parameters</th>
<th>Available parameters</th>
<th>Available parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agency</td>
<td>EffectiveDate</td>
<td>LastName</td>
</tr>
<tr>
<td>AgencyPhone</td>
<td>ExpirationDate</td>
<td>PolicyNumber</td>
</tr>
</tbody>
</table>
Setting up the Generate Letter step
The Generate Letter step is a component step. See the next section on how to set up a component step.

6.5.1 Setting up a component step

A component step calls Java methods that are available for use by the workflow process to complete a specific task. For our cases study, the Generate Letter step calls a Java method that copies a word document and another file to the claim folder of the current claim that is being processed.

To set up a component step:

1. Select the component step.
   
   For our case study, we select the Generate Letter step.

2. In the Properties pane on the right side, in the shaded Operations section, click Add (see Figure 6-37).

![Figure 6-37 Generate Letter - Define a component step]

3. From the component drop-down list, select the component to use.

   For our case study, we click More_Operations.

   **Note:** More_Operations is a component that has been made available for our case study. It is an unsupported IBM code at the time of writing.

   For complete steps in setting up a Component Integrator before you use it in your workflow process, refer to 8.2, “Implementing Component Integrator” on page 274.
4. From the Operation list, select one or more operations that are associated with the selected component and click **OK**.

For our case study, we select **copy** as shown in Figure 6-38.

![Operation Selection](image)

*Figure 6-38  Copy function from the More_Operations component*

5. For each operation, under the shaded Operation Parameters list, enter values for the parameters that are required for the operation.

For our case study, the following parameters are required for the copy operation:

- **Object**: A container for the document to be copied. In the Expression drop-down list, select **WordTemplate** as shown in Figure 6-39.
Figure 6-39  Copy function parameter set up: Object

- **File**: Specifies whether a copy of the document is to be created in the same folder as the old one. For the case study, set it to false by double-clicking the Expression box and entering `false`.

- **Copy**: The original document to be copied. From the Expression drop-down list, select `ClaimNotice`. 
6. Table 6-23 shows the component set up for the Generate Letter step. Using the table and set up the remaining CE_Operation component and its function.

<table>
<thead>
<tr>
<th>Component name</th>
<th>Operation names</th>
<th>Parameter names</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>More_Operations</td>
<td>Copy</td>
<td>Object</td>
<td>WordTemplate</td>
</tr>
<tr>
<td></td>
<td>File</td>
<td>File</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>Copy</td>
<td>Copy</td>
<td>ClaimNotice</td>
</tr>
<tr>
<td>CE_Operation</td>
<td>file</td>
<td>sourceDocument</td>
<td>ClaimNotice</td>
</tr>
<tr>
<td></td>
<td>destFolder</td>
<td>destFolder</td>
<td>CaseFolder</td>
</tr>
</tbody>
</table>

### 6.6 Building a submap

In this section, we show you how to build a submap. Submaps help you to simplify complex workflow processes. You can group related tasks and activities into logical modules and create corresponding submaps to accomplish these set of tasks and activities. You can reuse the submaps within your workflow or copy them to other workflow processes. Submaps can also call other submaps to further modularize works.

Building a submap is similar to building a normal workflow process. It consists of the following steps:

2. Add steps and routes as a normal workflow map.
3. Call this submap from other workflow process.

A submap cannot execute by itself. It must be called by the main workflow map or other submaps.

For our case study, we create the Claim Setup submap.

### 6.6.1 Creating the Claim Setup submap

For our case study, the Claim Setup submap handles the following tasks:

1. Create a claim folder based on a system-generated Claim ID.
2. Call the Get Field Agent step to assign a field agent to the claim.
3. Query Content Engine for all documents related to the claimant’s last name. We assume, to simplify the case study, that all of these documents with the claimant’s last name are related to the claim. In reality, we should use more fine-tuned criteria to search for related documents and add to the folder. This step actually does not add the documents to the folder. At the time of writing, we decide to search for the documents without adding them into the folder.

4. Wait until the assigned field agent adds one or more supporting documents to the claim folder. To simplify the case study, we assume that when at least one supporting document is added to the claim folder, the field agent’s job is done. In reality, we probably want to make sure the right documents (such as multiple vehicle repair estimates and the police report) have been received for the claim (that is added to the claim folder) before continuing to the next step.

5. Call the Get Adjustor step to assign the appropriate adjustor for the claim.

6. Complete its process and returns to the calling workflow map.

**Creating new submap**

To create the new Claim Setup submap:

1. From Process Designer, click the **Create Map** icon as shown in Figure 6-40.
2. A Create Map dialog box opens. Select **Create New Map**.

3. Enter a descriptive name for the Name field of the new map. For our case study, we enter **Claim Setup Submap**.

4. Click **OK**, see Figure 6-41.

![Create Map dialog box](image)

**Figure 6-41** Create Map dialog box

### Adding steps

For our case study, Table 6-24 lists the steps in the Claim Setup submap. Add these steps as shown in Figure 6-42.

<table>
<thead>
<tr>
<th>Step name</th>
<th>Step type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Case</td>
<td>Component</td>
</tr>
<tr>
<td>Retrieve Related Docs</td>
<td>Component</td>
</tr>
<tr>
<td>Get Field Agent</td>
<td>System</td>
</tr>
<tr>
<td>Field Agent</td>
<td>General</td>
</tr>
<tr>
<td>Get Adjustor</td>
<td>System</td>
</tr>
<tr>
<td>Check for Attachments</td>
<td>Component</td>
</tr>
<tr>
<td>File Supporting Docs</td>
<td>Component</td>
</tr>
</tbody>
</table>
Adding routes

Add routes to connect the steps in the Claim Setup submap as shown in Figure 6-43.
Defining conditional routing

In the case study, when a field agent attaches a document to the claim folder in the Field Agent step, the workflow property AttachmentCheck is set to true. The numbered routes, [1] from the Check for Attachments step to the Field Agent step, and [2] from the Check for Attachments step to the File Supporting Docs step, are conditional routes based on the value of AttachmentCheck.

At the Check Attachments step, we want to see if there is any attachment associated with the case study. If AttachmentCheck is set to false — that is, the field agent has not attached any document to the claim folder — then the workflow process goes back to the Field Agent step and waits for the field agent to attach related documents to the claim folder. If AttachmentCheck is set to true, that is there is at least one attachment associated with the claim folder, then the conditional routing goes to the File Supporting Docs step.

In real life, the conditional routing would be more complicated than what we show here. As mentioned earlier, depending on your business process requirements, you can set the Check for Attachments step to check whether a policy report has been filed and whether the vehicle repair estimate from an approved auto-repair shop has been submitted.

To set up the conditional routings for Claim Setup submap:

1. Select the route [1] from Check for Attachments to Field Agent.
2. From the Properties pane on the right side, click **Condition**.
3. Select the **Data Fields** tab.
4. Select **AttachmentCheck (Boolean)** from Field drop-down list.
5. Select **is equal** for Operator.
6. Enter **false** for Value.
7. Click the **Insert** button. See Figure 6-44.

Table 6-25 shows the conditional routing set up in the Claim Setup submap. Notice that the routing [1] from Check for Attachment to Field Agent is always checked first. If the condition is not true, the second routing [2] always executes.

**Table 6-25  Conditional routing in Claim Setup map**

<table>
<thead>
<tr>
<th>Source step</th>
<th>Target step</th>
<th>Conditional routing type</th>
<th>Data fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check for Attachment</td>
<td>Field Agent</td>
<td>Condition</td>
<td>AttachmentCheck = false</td>
</tr>
<tr>
<td>Check for Attachment</td>
<td>File Supporting Docs</td>
<td>Always true</td>
<td></td>
</tr>
</tbody>
</table>

**Setting up the Create Case step**

In the Claim Setup submap, there is a component step called Create Case. Table 6-26 shows the setup for the Create Case step.
### Table 6-26  Component and function setup for the Create Case step

<table>
<thead>
<tr>
<th>Component name</th>
<th>Function name</th>
<th>Parameter Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>More_Operations</td>
<td>createFolder</td>
<td>objectStoreName</td>
<td>This should be set to the object store you are using on your system. For our case study, it is &quot;objectstore&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>parentFolder</td>
<td>ParentFolder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>className</td>
<td>&quot;Folder&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>properties</td>
<td>{&quot;FolderName&quot;, &quot;STRING&quot;, ClaimID}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>folder</td>
<td>CaseFolder</td>
</tr>
<tr>
<td>More_Operations</td>
<td>searchForOne</td>
<td>searchObject</td>
<td>SearchPolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>objectType</td>
<td>&quot;document&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>itemIds</td>
<td>{1}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>values</td>
<td>{LastName}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>object</td>
<td>PolicyDoc</td>
</tr>
<tr>
<td>CE_Operations</td>
<td>file</td>
<td>sourceDocument</td>
<td>ClaimForm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>destFolder</td>
<td>CaseFolder</td>
</tr>
</tbody>
</table>

**Note:** It is important that the functions are added in the order listed. When control is passed to a component step, the functions are executed in a serial order. Also, remember the quotes around the expressions. In an expression, if the quotation marks are used, then the value of the parameter is set to the value within the quote. Finally, you must press Enter after each entry to ensure that the values are registered.

For details of how to set up a component step, refer to 6.5.1, “Setting up a component step” on page 169.

The descriptions of the parameters used in the More_Operations functions can be found in the Installation Guide for the More_Operations class. The description of the parameters for the CE_Operation class can be found in the product manual.
Setting up the Retrieve Related Docs step
The Retrieve Related Docs step is also a component step where we search for documents related to the claim. Table 6-26 shows the setup for the Retrieve Related Docs step.

Table 6-27 Component and function setup for the Retrieve Related Docs step

<table>
<thead>
<tr>
<th>Component name</th>
<th>Function name</th>
<th>Parameter Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>More_Operations</td>
<td>searchForMany</td>
<td>searchObject</td>
<td>SearchSupportingDocs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>objectType</td>
<td>&quot;document&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>itemIds</td>
<td>{1}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>values</td>
<td>{LastName}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>objects</td>
<td>SupportingDocs</td>
</tr>
</tbody>
</table>

Note: This step only searches for documents related to the claim but it does not do anything. To simplify the case study, we keep the step as is. In real life, we would probably want to search for the related documents and then add to the claim folder.

Setting up the Field Agent step
For our case study, we set up the Field Agent step as follows:

- The General tab:
  - Instruction: Please include your damage estimate and file all supporting documents before submitting.
  - Participants: Field_Agent workgroup
  - Participant Privileges: Clear all
- Parameters (see Table 6-28).

Table 6-28 Field Agent parameters

<table>
<thead>
<tr>
<th>Available parameters</th>
<th>Available parameters</th>
<th>Available parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccidentDescription</td>
<td>Company</td>
<td>LastName</td>
</tr>
<tr>
<td>AdjustorName</td>
<td>DebitAccountNumber</td>
<td>LossLocation</td>
</tr>
<tr>
<td>Agency</td>
<td>EffectiveDate</td>
<td>PolicyNumber</td>
</tr>
<tr>
<td>AgencyPhone</td>
<td>Email</td>
<td>ReportedDate</td>
</tr>
</tbody>
</table>
Assignments.

To simplify the case study, before the execution of the step, we assign SupportingDocs to an array of empty objects. See Table 6-29.

Table 6-29   Before execution tab

<table>
<thead>
<tr>
<th>Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>SupportingDocs</td>
<td>{&quot;&quot;}</td>
</tr>
</tbody>
</table>

After the execution of the step, we want to count the number of supporting documents that have been added to the claim case:

- If there is no support document added to the claim folder, which is `elementcount(SupportingDocs) = 0`, then:
  - Set the AttachmentCheck field to false.
  - Set F_Subject to the following text:
    "Please add the supporting documents to this claim: " + ClaimID

- Otherwise:
  - Set the AttachmentCheck field to true.
  - Set F_Subject to the following text:
    "New claim ClaimID: " + ClaimID

The AttachmentCheck field value (true or false) is used in the conditional routing to determine whether the control should go back to the Field Agent step again or move forward to the next step. If the controls are to go back to the Field Agent step, we also set the F_Subject value to inform the field agent at that step that supporting documents have to be added to the claim.
See Table 6-30 for the assignment set up after the step completion.

Table 6-30  After completion tab

<table>
<thead>
<tr>
<th>Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttachmentCheck</td>
<td>if(elementcount(SupportingDocs) = 0, false, true)</td>
</tr>
<tr>
<td>F_Subject</td>
<td>if(elementcount(SupportingDocs) = 0, &quot;Please add the supporting documents to this claim: &quot; + ClaimID, &quot;New claim ClaimID: &quot; + ClaimID)</td>
</tr>
</tbody>
</table>

Setting up the File Supporting Docs step

This is another component step where we use More_Operations to file all the supporting documents to the claim folder. Table 6-26 shows the setup for the File Supporting Docs step.

Table 6-31  Component and function setup for the File Supporting Docs step

<table>
<thead>
<tr>
<th>Component name</th>
<th>Function name</th>
<th>Parameter Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>More_Operations</td>
<td>fileMany</td>
<td>folder</td>
<td>CaseFolder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>objects</td>
<td>SupportingDocs</td>
</tr>
</tbody>
</table>

Setting up the Get Field Agent system step

The Get Field Agent step is a system step. Refer to the next section on how to set up a system step.

Setting up the Get Adjustor system step

The Get Adjustor step is a system step. Refer to the next section on how to set up the system step.

6.6.2 Setting up system steps

A system step enables you to call system functions provided by IBM FileNet Business Process Manager. Some of the system functions include assigning a data field in the workflow, launch another workflow process, and execute a procedure in a database.

For our case study, in the Claim Setup submap, we use two system steps:

- Get Field Agent
- Get Adjustor
Setting up the Get Field Agent step

For the case study, if the loss city of the claim is Los Angeles, then the field agent is assigned to Fran. Otherwise, the field agent is assigned to Fred. In real life, you want to make the assignment to a workgroup rather than a specific field agent name. In addition, you might want to obtain the field agent name from a database according to the loss location. For simplicity of the case study, we hard-code the condition.

To set up the Get Field Agent system step:

1. Make sure you select the Claim Setup submap from the drop-down list above the map window.

2. Select the Get Field Agent system step from the Claim Setup submap.

3. From the Properties pane on the right side, select Assign from the Available Functions list, add it to the Selected Functions list as shown in Figure 6-45.

4. Double-click Assign from the Selected functions and the Assign dialog box opens.

5. Select FieldAgent from the drop-down list as shown in Figure 6-46.
6. Click the Expression input box. Click the **Expression** icon and the Expression Builder dialog box opens.

7. Enter the following expression on the lower input box and click **OK**:
   
   ```java
   if(LossCity == "Los Angeles", "Fran", "Fred")
   ```

8. The Assign dialog box should look similar to Figure 6-47. Click **Close**.

   **Note:** Make sure to periodically save your work.
Setting up the Get Adjustor step
The Get Adjustor step is a system step. If the estimated vehicle repair amount (VehicleEstimateAmount) is less than or equal to 7000, assign Annie as the adjustor for the claim. Otherwise, assign Author as the adjustor.

This is the exact same setup as for the Get Field Agent step. Use the Assign function, and assign Adjustor with the following expression:

if(VehicleEstimateAmount <= 7000, "Annie", "Author")

See Figure 6-48 for the field assignment.

6.6.3 Linking the submap to the main map
To link a submap to the main workflow process map or calling map:

1. Select the calling map from the drop-down list as shown in Figure 6-49. For our case study, we select Workflow (Main Map).
2. Select the submap step on the main map that you want to link to a submap. In our case study, we select the **Claim Setup** step.

3. From the Properties pane on the right side, select the called submap from the Map drop-down list for your submap step. The drop-down list contains a list of submaps you created for this workflow process. For our case study, we select **Claim Setup Submap** from the drop-down list. See Figure 6-50.

4. To test the connection, double-click the submap step and the called submap should open. For our case study, click the **Claim Setup** step on the main map and the Claim Setup submap opens.

**Validating map**

To validate your work, select **File → Validate**. Make sure you resolve any issues before continue with the rest of the chapter.
6.6.4 Checking in and transferring your workflow process map

In 6.3.2, “Saving the workflow process” on page 142, we mentioned that there are several ways to save your workflow process. Before the workflow process can be launched, you have to perform the following steps:

1. **Check in** the workflow process to Content Engine.
2. **Transfer** the workflow process from Content Engine to the Process Engine.

To check in, use File → **IBM FileNet Add New** from Process Designer. For details, refer to “Option 2: IBM FileNet Add New” on page 143.

To transfer the workflow process, use File → **Transfer**.

6.7 Creating a workflow subscription

After you finish creating a workflow, you can create a workflow subscription that specifies when to launch the workflow based on events occurred on documents, folders, or custom objects. Events include creation and modification of documents, folders, or custom objects. For our case study, the event that launches the workflow is the creation of a Claim form. The Claim form is a custom object class.

**Note:** Before you create a workflow subscription, make sure that the associated workflow is saved, checked into Content Engine, and transferred to the Process Engine.
To create a workflow subscription, follow these steps:

1. From the workplace main menu, select **Author → Advanced Tools → Add Workflow Subscription** as shown in Figure 6-51.

![Add workflow subscription through workplace](image)
2. Select the target object class that launches the workflow:
   
a. Navigate through the object store to where the target object will be stored. For our case study, we select **objectstore**. See Figure 6-52.

![Figure 6-52  Add workflow subscription: Select target object store](image1)

b. Select the target object class type. For our case study, the object type of the Claim object class is a custom object. We select **Custom Object**. See Figure 6-53.

![Figure 6-53  Add workflow subscription: Select target object class type](image2)
c. Select the target object class and click **Next**. For our case study, the object class is called Basic Process. We select **Basic Process**. See Figure 6-54.

![Figure 6-54](image1)

Figure 6-54  Add workflow subscription: Select target object class

3. Select the workflow process that is launched by the subscription:
   a. On the left side of the window, the arrow should be placed next to the Select Workflow step. Click **Browse/Search for Workflow Definition**. See Figure 6-55.

![Figure 6-55](image2)

Figure 6-55  Add workflow subscription: Select workflow process, browse the folders
b. Navigate to the location where your workflow is stored in the object store. For our case study, we select **objectstore → Case Study Process**. See Figure 6-56.

![Figure 6-56 Add workflow subscription: Select the workflow process](image.jpg)
c. Click **Select from Versions** under your workflow process and select the version of the workflow that you want the workflow subscription to launch. For our case study, we click the link under XYZ Basic Process. See Figure 6-57.

![Image of workflow subscription selection](image)

**Figure 6-57**  Add workflow subscription: Select the workflow version to launch
d. Find the workflow version that you want the workflow subscription to launch and click Select under the version. For our case study, we select the latest version, 8.0, of our workflow. See Figure 6-58.

**Note:** In most cases, you select the most current workflow version. However, you have the option to select an older version of a workflow if required.

![Figure 6-58  Add workflow subscription: Select workflow version](image)

- Click **Next** after the workflow version has been selected.

4. Set the subscription properties:

a. Enter information for the following fields (see Figure 6-59):
   - Name: Enter a name for the workflow subscription. For our case study, we enter **XYZ Basic Process Subscription**.
   - Description: Enter an appropriate description for the subscription.
   - Initial State: Leave the check box selected.
   - Enable Manual Launch: Change it to **No**.
b. Click **Subscribed Events**.

c. Subscribe the events that launch a workflow. The available events for our case study’s target object class include:

- Change Class Event
- Creation Event
- Deletion Event
- Lock Event
- Unlock Event
- Update Event
- Update Security Event

There are different events available for subscription based on the target object class that you selected to launch a workflow.
To subscribe to events, select the entries from the Available Events list on the left side of the windows, and click the right arrow key to move them to the Selected Events list on the right side. For our case study, we select **Creation Event**. This means when our target object is created, the workflow is launched. See Figure 6-60.

![Select Subscribed Events](image)

*Figure 6-60 Select the creation event*

5. Click **Accept** and **Next** to move to the next step. See Figure 6-61.
6. Click **Next** to move on to the next pane. We do not build any special expressions for the workflow subscription. Click **Next** again to continue to the next step.

7. You should now be at the Set Property Map step. See the arrow on the left side of the pane as shown in Figure 6-62.

   In this step, you map the properties of your target object class to the data fields used of the workflow to be launched:

   a. From the left side, the Data Field Name drop-down box, select the data field of the workflow that requires data information. For our case study, we select **Agency (String)**. See Figure 6-62.
Introducing IBM FileNet Business Process Manager

Note: The data fields on the left side of the pane represent the data fields you define for the workflow (which is stored in the Process Engine). The properties on the right side of the pane represent the properties you define for your target object class (which is stored in Content Engine).

When you launch a workflow based on the subscribed event that occurred on a target object, the property information from the target object will be mapped to the data fields of the launched workflow. This way, the workflow gets the information it needs from the target object to do its work.

b. From the right side, the Property Name drop-down box, select the property of the target object class that triggers the workflow. For our case study, we select **Agency (String)**. See Figure 6-63.

c. Click the plus sign icon to complete the mapping. See Figure 6-64.

d. Repeat the previous three steps for other property mappings. When all data fields of the workflow are mapped by the properties of the target object class, click **Next**.
Figure 6-63  Set property map: Select the property name of the workflow to be launched

Figure 6-64  Set property map: Click the plus sign icon to complete the mapping
For our case study, we map the data fields of the target object class with the properties of the workflow as shown in Table 6-32.

Note: If you follow through our case study example, make sure that you map things precisely as described in the table.

Table 6-32  Property mapping for case study

<table>
<thead>
<tr>
<th>Data Field Name (of the workflow)</th>
<th>Property Name (of the target object class)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccidentDate</td>
<td>Date</td>
</tr>
<tr>
<td>Agency</td>
<td>Agency</td>
</tr>
<tr>
<td>AgentName</td>
<td>AgentName</td>
</tr>
<tr>
<td>ClaimID</td>
<td>ClaimID</td>
</tr>
<tr>
<td>Company</td>
<td>Company</td>
</tr>
<tr>
<td>Creator</td>
<td>Creator</td>
</tr>
<tr>
<td>EffectiveDate</td>
<td>EffectiveDate</td>
</tr>
<tr>
<td>LastName</td>
<td>LastName</td>
</tr>
<tr>
<td>LossLocation</td>
<td>LossLocation</td>
</tr>
<tr>
<td>PolicyNumber</td>
<td>PolicyNumber</td>
</tr>
<tr>
<td>VehicleEstimateAmount</td>
<td>VehicleEstimateAmount</td>
</tr>
<tr>
<td>VehicleReplacement</td>
<td>VehicleReplacement</td>
</tr>
</tbody>
</table>

The data fields of the workflow and the properties of the target object class might not always have a one-to-one naming correspondence. For our case study, the workflow uses a data field called AccidentDate. The object class uses a property called Date. We map Date from the object class to AccidentDate in the workflow.

8. The last step in the workflow subscription creation process includes setting the security. In this step, you can specify who has owner control, modify control, view control, and remove right of the workflow subscription. For our case study, we use the default.
9. Click **Finish** to complete the workflow subscription creation procedure. See Figure 6-65.

![Add workflow subscription: Click Finish to complete the subscription](image)

**Figure 6-65** Add workflow subscription: Click Finish to complete the subscription

**Note:** You can also create a workflow subscription using IBM FileNet Enterprise Manager if you want to create a workflow subscription for a particular document, folder, or custom object. For simplicity of discussion, we do not cover this topic in the book.

### 6.7.1 Exposing a system property and mapping it in a workflow

System properties cannot be mapped to the workflow’s data fields by default. However, there are times when the system properties have to be exposed to the launched workflow. For example, in our case study, we would like to capture the person that starts the workflow. This is the person who creates a target object that then triggers the workflow to be launched. The system property that contains this person's information is *Creator*. By default, it is not in the object class. This property needs to be exposed in the object class that we have built and use to launch the workflow.

If you imported the Content Engine objects from the downloadable materials provided with this book, the *Creator* property has already been exposed in the object class.
Because it is important to understand how to expose system properties in case you are required to do so, we outline the steps to accomplish this task here:

1. Start the IBM FileNet Enterprise Manager application:

   From Content Engine, select **Start → Programs → IBM FileNet P8 Platform → Enterprise Manager SnapIn**.

2. Navigate to the target object class.

   For our case study, the target object class we use is a custom object, called Basic Process, and it is stored in our object store called objectstore. Therefore, we select **Object Stores → objectstore → Other Classes → Custom Object → Basic Process**. See Figure 6-66.

![Figure 6-66 Browse to the target object class - For our case study, it is Basic Process](image)
3. Open the object class properties dialog box:
   Right-clicking your target object class and select Properties from the pop-up menu. For our case study, we right-click **Basic Process** and select **Properties**.

4. Expose the system property in the object class:
   a. Select the **Property Definitions** tab. See Figure 6-67.

![Figure 6-67  System properties definitions](image)

   b. Select the **System Properties** check box. This enables the system properties to be exposed for property mapping when creating a workflow subscription.

   c. Select the system property you want to expose for property mapping. For our case study, we select **Creator**.
d. Click **Edit**. The information for the system property displays (see Figure 6-68).

e. Find the symbolic name for the system property (in the Symbolic Name field). This is the name that you use for property mapping when creating a workflow subscription. It might not be the same name as what you see from the previous dialog box. Make a note of the symbolic name.

![Figure 6-68 Symbolic name](image)

f. Click **OK** to go back to the object class property dialog box.

5. Map the system property exposed earlier in the workflow subscription setting:

   a. Select the **Subscription** tab (from the object class properties dialog box).

   b. Select the workflow subscription you created earlier under the Subscription Name section. For our case study, we select Basic Process Subscription. See Figure 6-69.
c. Click **Properties** to show the workflow subscription properties. See Figure 6-70.
d. Select the **Workflow** tab as shown in Figure 6-71.

![Figure 6-71 Add system property to the subscription](image)

**Figure 6-71 Add system property to the subscription**

e. Add the system property that you want to expose here:

i. Click **Add**. The Property Map dialog box displays (see Figure 6-72).

ii. Enter the data field name of the workflow (Workflow Property) to be mapped to. For our case study, we enter **Creator**.

iii. Enter the system property name (Document Property) to be mapped. For our case study, we enter **Creator**.

iv. Click **Add**.

![Figure 6-72 Map the system to process property](image)

**Figure 6-72 Map the system to process property**
6. Click **OK** in the object class properties dialog box. For our case study, see Figure 6-73. Notice that the system property Creator is mapped in the workflow subscription.

![Figure 6-73  Complete exposing of the system property and its mapping to the workflow](image)

### 6.8 Testing the basic process

Always test your workflow processes when you have completed them. Make sure that you test the workflow subscription to ensure that the specified event does indeed launch the workflow.

### 6.8.1 Monitoring workflow processes

To test your workflow processes, we first describe two application tools that you can use to monitor your workflow processes:

- Process Tracker
- Process administrator
Process Tracker
Process Tracker provides the status of a workflow that is currently running in a graphical view. From the Process Tracker graphic view window, you can tell what steps have been completed in the workflow, when they were completed, and which steps are currently active.

Enabling users to use Process Tracker
To enable users to use Process Tracker (other than the Administrator), users must be added in the F_Trackers workflow group as follows:

1. Navigate to the workflow group setup:
   Open the workflow and go to **Workflow Properties → Workflow Groups**. See Figure 6-74.

![Figure 6-74 Add workflow group in a workflow property: F_Trackers](image)

2. Add participants to the F_Trackers workflow group. The participants of this workflow group will be allowed to track the workflow process from Workplace.

   **Note:** To save the time and effort of logging in and out as different users, this process is set up so that testing can be done using the administrator’s task box. The Administrator has not been added to the F_Trackers workgroup, because the Administrator has access to the Process Administrator, which can view Process Tracker. If you would like to add Administrator to the F_Trackers workgroup now, you have to save and transfer the new process and then modify your subscription to use the new version of the process.
Launching Process Tracker

To launch Process Tracker:

1. From Workplace, go to Tasks → Task Tracker (see Figure 6-75).
2. Select the workflow process you want to track.

![Figure 6-75 Workplace task tracker](image)

Using Process Tracker

Process Tracker displays a combination of information from the workflow definition, the values in the active work items, and the event logs. The Process Tracker's main window is divided into the following areas:

- The workflow map that shows the workflow's current status visually.
  
  The map shows the steps and routes defined for the workflow. The currently active work items are indicated with the blue arrow. Steps that have been traveled are indicated by the completed check marks. See Figure 6-76.
The workflow properties that show the details of the current status.

On the right side of the window, you can see the details of the selected item of a workflow, a step, or a route. See Figure 6-77.

Note: The cursor in the figure is pointing to a blue arrow. The blue arrow and the one above it allow you to expand either the workflow map or the properties pane to the full width of the window. These arrows can also be found on the Process Designer window.
To show the progress of a process, click the **Refresh** icon. See Figure 6-78.

Figure 6-77  Process tracker window properties pane

Figure 6-78  Refresh the view of the process
Introducing IBM FileNet Business Process Manager

**Process Administrator**

With Process Administrator, you can search for and view workflows, edit workflow data and properties, and manage workflows.

**Launching Process Administrator**

To launch the Process Administrator application, go to Admin → Process Administrator from Workplace. See Figure 6-79.

---

**Note:** The boxed number to the left of the Create Case step indicates the number of components that have successfully completed within that step (remember that there are three operations associated with this step in our case study).

Also in this view, the step currently being executed is the Field Agent step, indicated by the hour glass. This tells you that input from the field agent is required. It is waiting for the field agent to add supporting documents.
Using Process Administrator

Process Administrator provides a wide variety of options so you can focus your search very precisely. You can enter your search criteria and click Find Now to have the Process Administrator application find the workflow processes that you are looking for. See Figure 6-80.

![Process Administrator Search](image)

*Figure 6-80  Process administrator search.*

After you locate one or more workflows or work items (as shown in Figure 6-81, for example), you can use Process Administrator to make various changes to the work in progress, such as these:

- Complete a step and send it on to the next step.
- Modify workflow field values.
- Assign users to or remove users from a workflow group.
- Delete an entire workflow or one or more work items.
6.8.2 Testing the workflow process from beginning to end

To test the workflow process that you created, we recommend adding an object and triggers that are associated with the workflow and exercise through the entire workflow process from beginning to end. For our case study, we perform the following tasks to test the workflow process:

1. Create a new claim.
2. Add the supporting documents.
3. Complete the claim process.
4. View the completed claim.
Creating a new claim
For our case study, we test the workflow process by create a new claim:

1. From Workplace, navigate to your object store and the folder that contains your target object. For our case study, we select **Browse → objectstore → Claims**. See Figure 6-82.
2. Click **Add Custom Object** (see Figure 6-83).

![Figure 6-83   Add a new claim: Add a custom object](image)

3. Click **Change Class** (see Figure 6-84).

![Figure 6-84   Add a claim: Change target object class](image)
4. Select the target class. For our case study, we select **Basic Process** from the class list (see Figure 6-85).

![Figure 6-85 Add a claim: Select target object class](image-url)
5. Fill in the target object information. For our case study, we enter claim properties (see Figure 6-86). Click **Next**.

**Note:** The only required field is the **Claim ID**. It is required to create the claim folder.

*Figure 6-86  Add a claim: Enter claim information*
6. Accept the default security settings given (see Figure 6-87) and click **Finish**.

![Figure 6-87 Add a claim: Use default security and complete the claim](image)

7. Click **OK** on the Add Confirmation window (Figure 6-88).

   **Note:** The long name listed in the confirmation window is the internal ID for the object.

![Figure 6-88 Add a claim: Add confirmation window](image)
8. For our case study, when a claim is added, a new folder is created to hold the documents related to the claim. The new documents are also stored in the claim folder. See Figure 6-89.

Figure 6-89  New claim information icon
9. Located to the right of the new claim is an information icon. Clicking the information icon displays the data that is entered for the claim (see Figure 6-90).

![Figure 6-90  Claim data](image)

10. Click **Exit** to close the claim data window.
Adding supporting documents for the new claim
After the claim is added to the system, a field agent must review and claim and add any supporting documents to the claim folder. To test if this step works as expected, we walk through the adding of the supporting documents step:

1. From Workplace, click Tasks (see Figure 6-91).

Note: For this test, we have been logged in as Administrator so we can monitor the process with Process Tracker, and step the claim through the workflow process.

Figure 6-91 Look at the administrators tasks
2. Click on the claim we just entered (see Figure 6-92).

**Note:** If the new claim does not appear here, this might be for one of two reasons:

- The step that moves the new claim to the field agent has not completed yet. Wait a while for the step to be completed.
- There might be a problem with the workflow process definition. The problem can be seen in Process Tracker. Navigate from Workplace → Tasks → Task Tracker to see where the workflow process stopped.

3. Click **SupportingDocs** (see Figure 6-93 on page 222).
4. Click **Assign** (see Figure 6-94 on page 223).

**Note:** Documents can be attached to the claim in many ways. In our case study, we choose to attach the ones that reside in the content repository. The **Add New** link enables you to add documents outside of the repository.
Figure 6-93  Add supporting documents
Figure 6-94  Add document from repository
5. Click the repository that contains the document (see Figure 6-95).

![Image of Browse the object stores](image1)

Figure 6-95   Browse the object stores

6. Click the folder that contains document (see Figure 6-96).

![Image of Find the folder](image2)

Figure 6-96   Find the folder
7. Click the supporting document (see Figure 6-97).

![Figure 6-97](image)

Figure 6-97  Click the supporting document

8. Click **Complete** to move the claim to the next step in the workflow process (see Figure 6-98).
Introducing IBM FileNet Business Process Manager

Figure 6-98  Click complete
Completing the claim process

After completing the field agent review and adding the supporting documents to the claim folder, it reappears in the field agent’s (administrator) task box. The field agent has to complete the claim process to test the complete workflow process.

To complete the claim process:

1. From Workplace (as the field agent), go to Tasks → My Inbox.
2. Select the processed claim to view (see Figure 6-99).

![Figure 6-99   View claim in process](image)
3. Click **Complete** to complete the claim (see Figure 6-100).

![Image of Complete the claim](image)

**Figure 6-100  Complete the claim**

**Viewing the completed claim**

After the claim is processed, view the completed claim to double check that the entire workflow process works as expected:

1. From Workplace, navigate back to the claims folder to view the claim by **Browse → objectstore → claims** (see Figure 6-101).
Figure 6-101  Go to the claims folder

2. Select the new claim folder (see Figure 6-102).

Figure 6-102  Click the new claim folder

3. View the claim, the supporting documents, and the claim processed letter (see Figure 6-103).

Chapter 6. Implementing business processes: Case study  229
6.9 Creating and calling a Web service

Web services provide functions that can be used by other processes. There are many ways to create Web services. In this section, we show you how to create a Web service from a workflow process. In other words, we show you how to expose a workflow process such that it can be used as a Web service.

For our case study, we create two new workflow processes (GetAdjustor, and GetFieldAgent) and expose them as Web services. These new processes (Web services) replace the existing steps (Get Adjustor and Get Field Agent) in the basic workflow process.

**Note:** We cover the Web services concept in Chapter 7, “Integration with external systems and services with BPM” on page 257 and describe its complete implementation in Chapter 8, “Implementing Component Integrator and Web services” on page 271. If you are not familiar with Web services, or if you encounter problems while following the case study, we recommend reading these chapters first before you continue with the rest of the case study implementation.
6.9.1 Creating a workflow process as a Web service

Creating a workflow process to be used as a Web service consists of the following main steps:

1. Create a new workflow and define its workflow properties.
   Define the workflow properties as a normal workflow, except in the following two areas:
   - Data fields: These are the data fields that the workflow requires to process work. If this workflow process requires the calling process to pass in information, the parameters that contain the input information from the calling processes should be defined in the workflow properties definition.
   - Web services name and type: You must specify the name with which this workflow process is called by other processes. This name becomes the Web service name. Because this process is to be exposed as a Web service, it has to support Receive and Reply functions. Therefore, you have to set this process as a Receive/Reply Web service type in the workflow properties definition.

2. Create workflow steps and routes.
   Create the workflow steps and routes as a normal workflow. In addition, in order to enable this workflow process to be called as a Web service, the workflow process must contain at least one Receive step and optionally multiple Reply steps as follows:
   - Receive step: You must specify at least one Receive step in the workflow process. The operation parameters defined for the Receive step are the parameters provided by the calling process. The operation parameters should be defined in the data fields of the workflow properties.
   - Reply step: You can optionally include multiple Reply steps in the workflow process. The operation parameters and their values defined for the Reply step will be returned to the calling process.

3. Validate and check in the workflow process.

For our case study, we create two workflow processes that are to be used as Web services:

- GetFieldAgent
- GetAdjustor

Creating GetFieldAgent workflow process as a Web service
The GetFieldAgent workflow process duplicates the functions of the Get Field Agent step in the basic workflow process we defined earlier, but it will be used as the Web service that other processes call.
For simplicity of the workflow, we hard-coded the names of the field agents. The GetFieldAgent workflow process assigns either Fred or Fran as the field agent based on the location information provided by the calling process.

**Defining workflow properties**
To define workflow properties, follow these steps:

1. Start Process Designer:
   From Workplace, select **Author → Advanced Tools → Process Designer**.

2. Open the workflow properties:
   From the top menu, select **View → Workflow Properties**.

3. Select the **General** tab and enter the following information (see Figure 6-104):
   - Workflow Name: Enter **GetFieldAgent**.
   - Subject: Enter **GetFieldAgent**.

   ![Figure 6-104 GetFieldAgent workflow property - General tab](image)

4. Select the **Data Fields** tab and add the City and State fields as shown in Figure 6-105. When the other workflow process calls this Web service, the other workflow process should supply the City and State values.

   ![Figure 6-105 GetFieldAgent workflow property - Data Fields tab](image)
5. Select the **Web services** tab and then select the Partner Links tab. See Figure 6-106. This is where you define the Web service name and type.

![Figure 6-106 GetFieldAgent workflow property - Web services -> Partner Links tab](image)

6. Enter the information in the page as shown in Figure 6-107:
   - Double-click the cell under name in the Partner Links section, and type in **GetFieldAgent**.
   - Click the **Receive/Reply check** box.
   - Type in **GetFieldAgent** for the Process Port Type field.

7. Click **Close** to close the Workflow Properties window.

![Figure 6-107 GetFieldAgent workflow property - Web services -> Partner Links data](image)
**Adding the steps**

For our case study, add the steps of the workflow process as follows:

1. From the Process Designer map, select **Web services Palette** from the View Palette on the bottom of the window. See Figure 6-108.

![Figure 6-108 GetFieldAgent workflow map: Change to Web services Palette](image)

2. Place the steps on the workflow map as shown in Figure 6-109.

   The GetFieldAgent step is a Receive Web service step. The Fred and Fran steps are Reply Web services steps.

![Figure 6-109 GetFieldAgent workflow map: Draw the required steps](image)
Adding routes and setting up condition routing

Add the routes and set up the condition routings as follows:

1. Add routes as shown in Figure 6-110. Name the routes to Los Angeles and Denver as shown in the figure.

2. For the Denver route, if the City parameter is equal to “Denver”, the route goes to the Fran step. Set up the conditional routing for the Denver route as follows:
   a. Select Condition in the Properties pane.
   b. Select the **Data Fields** tab.
   c. Select **City (String)** from the Field drop-down list.
   d. Select **is equal** from the Operator drop-down list.
   e. Enter “Denver” in the Value field. Make sure to include the double quotes.
   f. Click **Insert**. Without clicking this button, the condition will not be saved.

3. For the Los Angeles route, if the City parameter is not equal to “Denver”, the route goes to the Fred step. Set up the conditional routing for the Los Angeles route as follows:
   a. Select **Condition** in the Properties pane.
   b. Select the **Data Fields** tab.
   c. Select **City (String)** from the Field drop-down list.
   d. Select **is not equal** from the Operator drop-down list.
   e. Enter “Denver” in the Value field. Make sure to include the double quotes.
   f. Click **Insert**. Without clicking this button, the condition will not be saved.
Introducing IBM FileNet Business Process Manager

Finishing defining the steps

After we add the steps and routes to the workflow map, we have to finish defining the details of each step:

1. Define the GetFieldAgent step:
   a. Select the GetFieldAgent step from the map and select GetFieldAgent from the Partner Link drop-down list as shown in Figure 6-111.

   ![Figure 6-111 GetFieldAgent workflow map: GetFieldAgent step - Partner Link property]

   **Note:** The system evaluates the routes in the order that they are created or in the order that you set up in the source step. If there are only two possible routes, for the route that will be evaluated at the last, you can select Always true without doing any special conditional routing setup.
b. Type in `GetFieldAgent` in the Operation field. See Figure 6-112.

![Figure 6-112 GetFieldAgent workflow map: GetFieldAgent step - Operation property](image)

Figure 6-112   GetFieldAgent workflow map: GetFieldAgent step - Operation property

c. Under the shaded Operation Parameters, specify the input parameter passed from the calling process (see Figure 6-113):
   i. Click the first cell under Name, and type in `City` and press Enter.
   ii. Select `String` from the Type drop-down list.
   iii. Select `City` from the Field Name drop-down list. If there is nothing in the drop-down list, type `City` and press Enter.
   iv. Repeat the same steps for the State parameter.

This workflow process takes the City and State information from the calling process and assigns a field agent based on the information. For simplification, although we pass both fields into the workflow process, we use just the City field when assigning a field agent.
2. Define the Fred step:
   a. Select the **General** tab. Select **GetFieldAgent** from the Partner Link drop-down list and select **GetFieldAgent** from the Operation drop-down list (see Figure 6-114).

![Figure 6-113  GetFieldAgent workflow map: GetFieldAgent step - Operation parameters](image)

![Figure 6-114  GetFieldAgent workflow map: Fred step - Operation property](image)
b. Under the Operation Parameters, enter FieldAgent for the Name field, select String from the Type field, and enter “Fred” in the Expression field. Remember to include the double quotes around Fred. See Figure 6-115.

![Figure 6-115  GetFieldAgent workflow map: Fred step - Operation parameters](image)

3. Define the Fran step using the similar procedures as the Fred step. Instead of “Fred”, type “Fran” in the Expression field under the Operation Parameters section.

**Validating workflow process map**
Using one of the validation methods we described earlier, validate the GetFieldAgent workflow process. There should be no errors. If you receive any errors, use the error messages to find and correct the problem.

**Reminder:** To validate a workflow process using the menu commands, select File → Validate.

**Checking in the workflow process**
After you validate that the workflow process has no error, transfer and check in the workflow process. See Figure 6-116. The transfer operation makes the process available for use.

**Reminder:** To transfer a workflow process using the menu commands, select File → Transfer.
Creating GetAdjustor workflow process as a Web service
The GetAdjustor workflow process duplicates the functions of Get Adjustor step in the basic workflow process we defined earlier, but it is to be used as the Web service that other processes call.

For simplicity of the workflow, we hard-coded the names of the adjustors. The GetAdjustor workflow process assigns Author or Annie as the adjustor name based on the vehicle’s estimated repair amount provided by the calling process.

Defining workflow properties
To define the workflow properties, follow these steps:
1. Select File → New Workflow.
2. Select View → Workflow Properties.
3. In the General tab, type GetAdjustor for both Workflow Name field and the Subject field. See Figure 6-117.
4. In the **Data Fields** tab, add the data field **Amount**, with type as Float. See Figure 6-118. When other workflow process calls this Web service, the other workflow process should supply the **Amount** value for this Web service.

5. In the **Web services** tab, go to the **Partner Links** tab. Enter the Partner Links information as shown in Figure 6-119. Note that we set the Web service with the name, **GetAdjustor**, of type **Receive/Reply**. This means that the Web service contains both Receive and Reply functions.

**Adding the steps and their routes**

For our case study, we add the steps and routes in the workflow map as shown in Figure 6-120. The GetAdjustor step is the Receive step. The Author and Annie
steps are the Reply steps. The route that connects GetAdjustor to Author is labeled >7K. the route that connects GetAdjustor to Annie is labeled as <=7K.

Finish defining the steps and routes
For our case study, both the Author and Annie steps are Reply steps. Each step sets the AdjustorName value. Based on the conditional routing, only one of the steps is executed and returns the AdjustorName value to the calling process.

To finish defining the steps for our case study:
1. Define the Author step:
   a. Select the Arthur step from the workflow map.
   b. In the General tab, fill out the information as shown in Figure 6-121.
2. Define the Annie step as shown in Figure 6-122.

![Figure 6-122](image)

Figure 6-122  GetAdjustor workflow map: Annie step properties

3. Define the >7K route as shown in Figure 6-123.

![Figure 6-123](image)

Figure 6-123  GetAdjustor workflow map: >7K route properties
4. Define the <=7K route as shown in Figure 6-124.

![Image of workflow process properties]

**Figure 6-124  GetAdjustor workflow map: <=7K route properties**

**Validating the workflow process map**
Using one of the validation methods we described earlier to validate the workflow process map. If you get an error, use the error message to correct the problem.

**Checking in the workflow process**
After validating the workflow process, transfer and check in the workflow process.

### 6.9.2 Calling the Web service in a workflow process

After you create a workflow process as a Web service, you can call the Web service from other workflows.

Calling a Web service from a workflow process consists of the following main steps:

1. Open the existing workflow and define its workflow properties:
   - Web service name and type - You must specify the Web service that the current workflow process will call (invoke) within its process. In addition, the specified Web service must be set up as an Invoke Web service type.
2. Add the Invoke step and routes:
   - Outgoing Parameters: If the Web service that the workflow process calls requires input values, the workflow process must provide the information in the Outgoing Parameters field. This corresponds to the operation parameters of the Receive step in the invoked Web service (workflow process).
   - Incoming Parameters: If the Web service that the workflow process calls returns values, the workflow process can get the returned information from these parameters. This corresponds to the operation parameters of the Reply steps in the invoked Web service (workflow process).
   - Routes and other steps: Create routes and other steps as necessary for the workflow.

3. Validate and check in the workflow process.

For our case study, we go back to our basic workflow process and update it to use the two new Web services we created in the previous section.

**Defining workflow properties**

Before using the existing Web services in a workflow map, you must define the Web services that you can invoke in the workflow properties:

1. Open the Claim Setup submap that needs to call the Web services.
2. Go to View → Workflow Properties → Web services → Partner Links.
3. Type GetFieldAgent in the Name field. This is the Web service that the current workflow process will call.
4. Check the **Invoke** box and click the **ellipse** button (see Figure 6-125).

![Figure 6-125  Claim Setup submap: Web services, Partner Links](image-url)
5. Fill in the information as shown in Figure 6-126:
   a. Click the **Web services workflows** radio button.
   b. Type `get` in the Name field.
   c. Click **Execute**.

   This gets a list of available Web services that the current workflow process can call.

![Web Services Explorer](image)

*Figure 6-126  Claim Setup submap: Browse for the Web services*
6. A list of Web services workflow displays. Select the Web service that your workflow invokes. For our case study, we select GetFieldAgent from the choices. Click OK (see Figure 6-127).

![Web Services Explorer](image)

Figure 6-127  Claim Setup submap: Select the GetFieldAgent service

**Deleting the existing steps (optional)**

If you have steps in your workflow that will be replaced by the Web services, delete these steps first from the workflow. For the case study, we delete the Get Field Agent and Get Adjustor steps.
To delete the obsolete steps in our case study:

1. From the workflow map, right-click the **Get Field Agent** step and select **Delete** from the pop-up menu (see Figure 6-128).

![Figure 6-128 Claim Setup submap: Delete obsoleted steps](image)

2. Repeat the delete procedure for the Get Adjustor step.
**Adding Web services steps in the workflow process**

To call the Web services in your workflow:

1. Change the View Palette to the Web services Palette (see Figure 6-129).

![Figure 6-129 Claim Setup submap: Change to Web services Palette](image)

2. Add the Invoke steps and routes in your workflow. For our case study, we add two Invoke steps: Get Field Agent and Get Adjustor steps (see Figure 6-130).
Introducing IBM FileNet Business Process Manager

Figure 6-130  Claim Setup submap: Calling Invoke steps

**Finishing defining the steps and routes**

To define the Invoke steps:

1. Define Get Field Agent step. The is an Invoke step that calls GetFieldAgent Web service. It requires location information as input. The values come from the workflow's City and State data field. Define the step as follows:
   a. Select the **Get Field Agent** step and click the **General** tab.
   b. Select **GetFieldAgent** from the Partner Link drop-down list (see Figure 6-131). This is where you specify what Web service you want to invoke.

Figure 6-131  Calling GetFieldAgent service: Partner Link
c. Select **GetFieldAgent** from the Operation drop-down list (see Figure 6-132). Some Web services have multiple operations. You need to specify which operation to invoke. For the Web services that we created earlier, there is only one corresponding operation per Web service.

*Figure 6-132  Calling GetFieldAgent service: Operation*
d. Ensure that the **Parameters** radio button is checked (see Figure 6-133).

e. Under the Outgoing Parameters section, set the values (in the Expression fields) for the parameters to be used when invoking the Web service. For our case study, the Web service requires the input values of City and State to determine which field agent to assign to the insurance claim case. Set these values as follows:

i. For the City parameter, from the Expression drop-down list, select **LossCity** (see Figure 6-133).

ii. For the State parameter, select **LossState**.

![Figure 6-133  Calling GetFieldAgent service: Outgoing Parameters](image-url)
f. Under the Incoming Parameters section, set the returning parameter names and values from the invoked Web service. For our case study, select **FieldAgent** for the Name and Field Name fields. See Figure 6-134.

![Figure 6-134](image)

**Figure 6-134** Calling GetFieldAgent service: Incoming Parameters

2. Define the Get Adjustor step. The step is an Invoke step that calls the GetAdjustor Web service. It requires the Amount parameter as input. Its value comes from the workflow’s VehicleEstimateAmount data field. Define the step as follows:

   a. Select the **Get Adjustor** step.

   b. In the **General** tab:

      i. Select **GetAdjustor** from the Partner Link drop-down list (see Figure 6-135).

      ii. Select **GetAdjustor** from the Operation drop-down list.
c. Ensure that the **Parameters** radio button is checked. See Figure 6-136.

d. Under the shaded Outgoing Parameters section, for the Amount parameter, from the Expression drop-down list, select **VehicleEstimateAmount**. This is how you set the workflow's data field value (from VehicleEstimateAmount) to the input parameter's value (Amount) required by the Web service.
e. Under the shaded Incoming Parameters section, for the AdjustorName parameter, from the Field Name drop-down list, select AdjustorName. See Figure 6-137. This is where the invoked Web service returns its values.

**Figure 6-137  Calling GetAdjustor service: Incoming Parameters**

**Check in and transfer your process**

Always check in the process before you can use it. For our case, we check in the process as XYZ Basic Process WS.
6.9.3 Create a subscription to launch your process

A new subscription needs to be created for the workflow process that you just modified. Use the following data to create the new subscription:

- Class - Basic Process WS
- Process - XYZ Basic Process WS

**Note:** Remember to select the most current version of the process.

**Exposing system data properties**

The Creator system property needs to be exposed in the custom class that we have built and uses to launch the workflow. For procedures on exposing the system properties, refer to 6.7.1, “Exposing a system property and mapping it in a workflow” on page 199.

6.9.4 Testing the process with Web services

Test the process as you did before. The only difference is that for this process, the object class you add for your new claim is **Basic Process WS**.
Chapter 7. Integration with external systems and services with BPM

In this chapter we introduce, from a high level business perspective, how IBM FileNet Business Process Manager system integrates with external systems and services.

We cover the following topics:
- Component Integrator
- Web services
- Rules connectivity framework

Next, in Chapter 8, “Implementing Component Integrator and Web services” on page 271, written from a technical perspective, we provide more information to help you understand and implement the Component Integrator and Web services that are used to integrate with external system and services.

For step-by-step instructions on how to implement these features for our case study, refer back to Chapter 6, “Implementing business processes: Case study” on page 127.
7.1 Component Integrator

*Component Integrator* enables the Process Engine to connect with external systems. With Component Integrator, you can use custom Java codes or existing JMS components, and make them available in a workflow.

JMS provides basic Enterprise Information System (EIS) integration. For JMS information, see:

http://java.sun.com/products/jms

Component Integrator provides extensible Java and JMS framework to enable IBM FileNet Business Process Manager to interact with any system. In addition, there is a component, *CE_Operations*, that allows access to Content Engine objects and properties from a workflow.

**How it works**

In Figure 7-1, a work item represents a workflow step waiting in the Process Engine for processing. The work item makes a request of a component. The component provides a response by performing some actions that might be outside of the Process Engine. The response is used in the subsequent steps on the workflow. You do not have to know the details of the component itself, it can be some custom Java code that connects to a legacy system, database server, or mail server. The advantage of Component Integrator is that it hides the complexity of the outside component.

![Figure 7-1  How Component Integrator works](image)
Component Integrator use case examples

In a typical scenario where Component Integrator might be used, a company has an existing legacy system and wants to leverage the functions provided by the system within a business process.

Following are some examples of how the Component Integrator can be used in an IBM FileNet Business Process Manager application:

- A company has existing Java codes that query proprietary systems for claimant policy information. Adjustors need this claimant policy information to adjust the claim amount in an IBM FileNet Business Process Manager workflow process. The Java codes can be made available through Component Integrator in the workflow step.

- A company has a Java application that stores and retrieves customer profile data from a legacy data store. In an IBM FileNet Business Process Manager workflow process, there is a requirement to retrieve and update customer profiles. Component Integrator can be used to utilize the existing Java codes for this IBM FileNet Business Process Manager workflow process.

- A company has a supplier that accepts order requests as XML documents through guaranteed messaging service using message queues, and wants to automatically generate order requests as part of its business process.

- An accountant uses existing JMS queue to send payment data. This data is used (also considered as being consumed) by an electronic payment system to process and issue check to customers.

- A customer has existing Java code that update sales order records in a proprietary sales order system, and wants to integrate these functions in an IBM FileNet Business Process Manager business process.

7.1.1 Content Engine operations

To provide content management operations in an object store, IBM FileNet uses the CE_Operations component to support Content Engine operations in workflow component steps.

The CE_Operations component provides the following functions:

- File a specific document to a given folder in the object store.
- Unfile a specific document from a given folder in the object store.
- Get the property value of a given document or folder.
- Set the property value of a given document or folder.
See Figure 7-2, which shows some of the CE_Operations you can select within a workflow process.

![Operation Selection](image1.png)

**Figure 7-2  CE_Operations**

Additional operations are offered through other components such as More_Operations and BPF_Operations.

The More_Operations component offers a rich set of functions (see Figure 7-3), some of which include:

- Create, delete, copy, and move documents, folders, or custom objects.
- Get and set multiple property values of a given document or folder.
- Check in and check out of a document.
- Search for a document or documents using search templates.
- Send e-mails with attachments, and optionally using templates.

![Operation Selection](image2.png)

**Figure 7-3  More_Operations**
7.2 Web services

Web services provide universal inter-operability between heterogeneous applications using Web standards.

The Process Engine uses Web services technology to enable interaction between IBM FileNet Business Process Manager processes, as well as between IBM FileNet Business Process Manager processes and other business processes.

There are three ways that IBM FileNet Business Process Manager uses the Web services technology:

- Invoke Web services from a BPM workflow process. This enables you to use the Web services developed for other applications in your organization and the ones that are available externally on the Internet.
- Provide a BPM workflow process as a Web service and enable other applications to use this BPM workflow process to accomplish their tasks.
- Use the Web service API to write step processors (not covered in this book).

There are two message type options when invoking or creating a Web service:

- Parameters: This is an easy way to use Web services. You do not need to understand XML or XML schemas.
- XML: This is for working with complex parameters that the Parameters message type cannot handle. To use it, you need to know XML and XML schema knowledge.

Web services standards

In order for you to understand how to use XML Web services within IBM FileNet P8, you should understand some basic definition of XML Web services.

There is a set of protocols that define how a Web service can be published, discovered, and used in a technology neutral, standard format. Among these standards are:

- Web Services Description Language (WSDL): Interface definition used to define Web services. A WSDL definition describes how to access a Web service and what operations it performs. You can use WSDL to create Web services and publish these services to UDDI.
- Universal Description, Discovery and Integration (UDDI): The directory that provides the means for discovering and publishing Web services. It is provided by a third party that makes these services available for inquiry. It is the equivalent of “Yellow and White Pages” of Web services.
Extensible Markup Language (XML): The base meta-language for many Service Oriented Architecture related technologies and standards. In Web services, the messages including embedded data are written in XML format. It can be used to exchange data, so that data can go back and forth between the services provider and the client in the standard recognizable format. You can consider it as the payload (in relation to HTTP).

Hypertext Transfer Protocol (HTTP): The vehicle that carries the XML data between systems. You can consider this as the transport.

Web services use case examples
Figure 7-4 illustrates a basic example of a Web services use case with three steps.

In the typical scenario, Company A wants to offer a service to other companies. Company A publishes the service as a Web service in a public UDDI repository (denoted as step 1). The UDDI registry entry includes the WSDL, which describes the service and how to use it.

Company B requires a particular service. Company B searches the public UDDI repository (denoted as step 2) and finds Company A's Web services offering, which meets its needs. Company B obtains the business contact information and WSDL from the UDDI registry entry, and uses (consumes) Company A's Web service (denoted as step 3).

An example of a service that Company A offers to other companies is credit score reporting (FICO Score). Company A generates revenue by offering this service to other companies through Web services. Company B, on the other hand, might be a financial institution that provide loans to customers. In order for Company B to process customer loans, company B uses (consumes) Company A's credit score reporting Web service to help determine whether to approve or reject the loan.
7.2.1 Process orchestration

Process orchestration describes the dynamic execution of a business process from a technical view\(^1\). In the IBM FileNet Business Process Manager environment, it describes the ability for processes to *use and provide* (consume and be consumed by) Web services. An executable process can use other processes to accomplish its work. It can also provide itself as a Web service for other processes to use. Process orchestration involves combining disparate components into joined-up process flows that automates end-to-end business functions.

Process orchestration is a key BPM market requirement because it:

- Enables end-to-end processes.
- Enables collaborative business process.
- Enables Business to Business (B2B) transactions.
- Enables Web services composition.
- Complements Web services.

---

\(^1\) Definition is extracted from [http://www.serviceoriented.org/process_orchestration.html](http://www.serviceoriented.org/process_orchestration.html)
IBM FileNet Business Process Manager processes can use (consume) existing Web services and provide (be consumed as) Web services. The IBM FileNet Business Process Manager processes can be published as Web services, which then can be discovered through a UDDI registry.

IBM FileNet Business Process Manager offers the following Web services system functions to achieve Process orchestration:

- **Invoke**: Requests a service from a selected partner link.
- **Receive**: Provides a service in response to a Web service request.
- **Reply**: Sends a response to a Web service associated with a previously accepted receive system function.

**Invoke function**

You can use the *Invoke* system function anywhere in a workflow process. The Invoke system function calls an application component that provides a Web service. This enables a workflow process to interact with and use external Web services.

Invoke is implemented as an operation on a Web services queue called *WSRequest*. When the IBM FileNet Business Process Manager system encounters an Invoke system function at a workflow step, the work item is routed to the WSRequest queue.

**Invoke example: XYZ Corporation Authorize Credit Web services**

Figure 7-5 illustrates an example of using the Invoke system function. The workflow process shown in Figure 7-5 uses an external Web services provided by Visa.com to authorize credit card transaction.

In the workflow process, there is a workflow step called Authorize Credit Card. At this step, an Invoke system function calls a Web service from a Web services provider, Visa.com. The Web service provides a credit authorization service. Authorization request information is sent to the Web services. The Web service returns a response, either Valid or Not Valid, back to the service requester (the Authorize Credit Card workflow step) in the workflow process. The returned response is evaluated at the next step of the workflow process and then routed to the appropriate steps to accept or deny the credit card transaction.

In this example, Visa.com is the Web services provider and XYZ Corporation's workflow process is the Web services consumer.
Figure 7-5  Web services

**Receive/Reply function**

Handling of inbound requests from external Web services is done using the Receive and Reply system functions.

The *Receive* system function can be viewed as representing an entry point to the workflow process. Incoming messages are processed by a listener called *WS-Listener*, which is running as part of Component Integrator. If you put a Receive step immediately after a Launch step in a workflow, an appropriate incoming message received by the system can launch a workflow process.

The *Reply* system function provides the ability to pass back information to the application that made the original request. There might be multiple Reply steps in a workflow process for one Receive step. The relationship between Receive and Reply is usually one to many.

Similar to the Invoke system function, the Reply system function is implemented as an operation on the *WSRequest* queue. When the Process Engine encounters a Reply system function in a workflow, the work item is routed to the WSRequest queue. The *WS-Adapter*, running under the Component Integrator, queries the queue and processes the work item.

Unlike the Invoke system function, there is no response received back from the call.
Receive/Reply example: Publish XYZ Corporation statistics

An example of using Receive and Reply system functions is in the insurance claims processing workflow that provides statistics lookup service to the executives of the company.

The executives require high level statistics on insurance claim totals and processing time displayed in their home page.

The IT department creates a workflow that collects current statistics from the claims processing database on demand. The workflow contains a Receive system step that gets the request for high level statistics. When the request is received, it launches the workflow process. A DB Lookup system step is called within the process that retrieves the current statistics from the Claims database, and summarizes them to the high level that is required by the executives. The workflow also contains a Reply system step that sends back the high level statistics information to the requester.

The IT department publishes the workflow to a UDDI registry. It also modifies the executives’ home page template to include the WSDL that associated with the Web services. The executives can use the link provided in their home page to get the high level statistics they need.

Figure 7-6 illustrates the example. Notice that, in this case, the workflow process is the Web services provider, and the executive home page is the Web services consumer.
7.3 Rules connectivity framework

The Rules Connectivity Framework (RCF) allows you to implement dynamic business rules for workflows. These rules can be simple or complex and can perform a variety of actions, such as performing if-and-then tests, setting values in data fields, or sending a work item to a submap or exception map. When you change the business rules, the changes are immediately applied to the existing workflow process.

**Note:** IBM FileNet Business Process Manager by itself does not provide a business rules engine. It integrates with the third-party rules engine software of your choice in its system.

Workflow process designer and business analysts can create and add business rules to individual steps of a workflow. Rules are used to separate business rules from the process, making it easier for a business analyst to adjust a complex process by modifying the rules rather than modifying a workflow process.
Rules can control the following types of operations in a workflow:

- Assign data to a field in the work item.
- Send a work item to a workflow map.
- Send a work item to a custom exception workflow map.
- Skip a workflow step.
- Repeat a workflow step.

To implement rules in a workflow process, the workflow designer and the business analyst work together to determine how rules should be used in the workflow, what business decisions should be controlled by the third-party rules engine, what workflow data are required for the rules engine, the appropriate names for the rule sets, and the steps in a workflow where the rules are executed. Each step in a workflow can have multiple rules, and the rules can be executed at different stages of the step, before execution, on save, or after completion.

**How it works**

When a workflow encounters a rule, the Process Engine sends a request to the *Rules Listener* (an interface provided by IBM FileNet), which passes the request to the rules engine. The rules engine executes the rule-set and returns the results to the Process Engine.

In IBM FileNet Business Process Manager environment, if the Rules Listener is stopped when a workflow tries to run a rule, the Process Engine sends the work item to a special queue called *Instruction Sheet Interpreter* (ISI). When the Rules Listener is restarted, the ISI will resubmit the work item to the Rules Listener.

If the rules engine interface is disabled in Process Configuration Console, the workflow will run and ignore any rules implemented in the workflow.

**Asynchronous rules processing**

Although the rules engine can execute most rule-sets in less than a second, some rule-sets, such as database lookups, might require longer processing time. When a Process Engine process (VWKs) is waiting for a response from the rules engine, it is not available to process any other incoming requests. To improve system performance, use asynchronous rules processing (executing rules in the background). You can configure specific rule-sets to be processed this way.
Overview of implementing rules

Business rules are assigned to workflows using Process Designer. To implement rules in your IBM FileNet Business Process Manager system, the following steps are required:

1. Install a Rules Listener and the third-party rules software on a server.
2. Enable rules in the Process Engine (through Process Configuration Console.)
4. Start and stop the Rules Listener (through Process Task Manager).

It is beyond the scope of this book to cover detailed steps of setting up a rules engine and using the rules. However, in the following two chapters, we provide more details on understanding and implementing Component Integrator.
Implementing Component Integrator and Web services

Previously, in Chapter 7, “Integration with external systems and services with BPM” on page 257, we introduced, from a high level business perspective, how IBM FileNet Business Process Manager integrates with external system and services.

In the current chapter, from a technical perspective, we provide information about understanding and implementing various integration options.

We cover the following topics:

- Understanding Component Integrator
- Implementing Component Integrator
- Understanding Web services
- Implementing a process to invoke a Web service
- Implementing a process as a Web service

For step-by-step instructions about how to implement Component Integrator and Web services for our case study, refer back to Chapter 6, “Implementing business processes: Case study” on page 127.
The Component Integrator is an IBM FileNet Business Process Manager service that connects a workflow with an external system. The Component Integrator provides an easy way to use external components such as Java classes in a workflow. You can use the components without programming. It is installed as part of a typical Application Engine server installation.

The Component Integrator consists of the following components:

- Component adaptors in the Application Engine server
- Component queues in the Process Engine server
- Component Manager in the Application Engine server

Component adaptors are used to interact with different types of components from a workflow step. You can use the Java adaptor to call Java components, and the JMS adaptor to interact with message queues. You can also write your own adaptors for different components and applications.

Component queues are created to service different types of component adaptors.

When a workflow process is executed, the Component Manager (Component Integrator Service Manager) retrieves the request from the component queues and invokes the components via the component adapters.

For each step in the workflow, the general sequence of events takes place as shown in Figure 8-1. The diagram illustrates the runtime interaction of the Component Integrator with Application Engine services (such as Component Manager), component queues in the Process Engine server, and a custom entity. The numbers in the diagram represent the steps in the operation of the Component Integrator.

Figure 8-1 shows the following flow:

- Information is sent to the component (via the adapter).
- The component performs its work and interacts with the custom entity.
- The result of the work is saved in the step (in the workflow).
- The step (in the workflow) is completed.
8.1.1 Adapter and security

In Component Integrator operations, access to the installed components is controlled through Java Authentication and Authorization Service (JAAS). It allows fine-grained security control over who is running code, as opposed to where the code comes from or who signed it.

JAAS can be used for two purposes:

- **Authentication**: To authenticate users to reliably and securely determine who is currently executing the Java code, regardless of whether the code is running as an application, an applet, a bean, or a servlet.

- **Authorization**: To ensure that users have the access control rights (permissions) required to do the actions performed.

JAAS is used for the Process Engine and component authentication of users. It supports a single sign-on capability.

For information about JAAS information, see:

http://java.sun.com/products/jaas
8.2 Implementing Component Integrator

The main steps to implement Component Integrator can be summarized as follows:

1. Create and configure a component queue.
2. Import component queue operations.
3. Configure Component Manager.
4. Use components in a workflow.

Various tools and applications are used to implement Component Integrator:

► **Process Configuration Console**: This is used to register the components. From the tool, you import service definitions for the component. You must create a component queue for each imported class and make the component queue operations available to use during a workflow design.

► **Process Designer**: This is used to define component steps in a workflow. When defining a component step, you select the component, the component operation to perform at the step, and provide workflow fields as parameters for the operation.

► **Process Task Manager**: This is used to start, stop, and administer Component Managers. Your component must be registered with a Component Manager. This can be verified under the Required Libraries tab for the Component Manager.

► **Component Manager**: This polls the component queues and retrieves work items that are dispatched to them. Component Manager passes a given work item to the adaptor that is assigned to the work item’s queue.

There is also a configuration option to allow the system to automatically check for new events. This option causes the Component Manager to respond to items in a component queue as they arrive. This event-driven mechanism can be enabled in addition to polling for work items.

If your IBM FileNet Business Process Manager system has been configured to automatically check for new events, the Component Manager responds to both the component’s polling rate and to new events as they occur.

8.2.1 Creating and configuring a component queue

Using Process Configuration Console, you can create and configure a component queue with a Java adaptor or a JMS adaptor. Use Add Component Queue Wizard page to associate Java adaptor with a Java application, and JMS queue with a message queue.
To create a new component queue (for example, More_Operations) in the Process Configuration Console tool, follow these steps:

1. Launch the Process Configuration Console tool.
2. Connect the isolated region (right-click the isolated region and select Connect from the context menu).
3. If there is no XML file provided with the new component, right-click Component Queues and select New from the context menu. See Figure 8-2. If an XML file is already provided, you can right-click Component Queues and select Import from XML file. Select the XML file, import it to the system. Afterwards, right-click the new component (for example, More_Operations) and select Properties. Then jump to step number 5 on page 277. The XML file might contain descriptions of the methods available in the Java class. This helps you to select the right methods to use later on in your workflow. Otherwise, you would have to either guess what they do, or manually add the descriptions for each method.

The user you used to log into the Workplace must have administrative rights on your Process Engine server.

Note:

To launch Process Configuration Console from Workplace XT:

1. Launch WorkplaceXT:
   
   
   
   http://<Application Engine server name>:9080/WorkplaceXT

2. Log in.

To launch Process Configuration Console from Workplace:

1. Launch Workplace:

   http://<Application Engine server name>:9080/Workplace

2. Log in.
4. The Add Component Queue Wizard starts. Enter the following information and click Next:
   - Queue Name: The name of the component queue. This name will appear in the list of component queues in Process Designer. For our example, we enter More_Operations.
   - Description: You can optionally enter a brief description of the queue. For More_Operations, we leave it blank. See Figure 8-3.
f. Select **Java Component** or **JMS Component** from the Adaptor drop-down list, and click **Configure**.

5. Configure the component:
   - For Java component, set information for the following fields and click **OK**:
     - **Jar File Path**: Locate and select the Jar file containing your Java component. For our More_Operations example, we select:
       c:\temp\more_operations\More_Operations.jar
     - **Java Class**: Select the appropriate class from the drop-down list of the available classes. After you select the Java class, a list of available methods display in the Available Methods box. When you configure the component queue operations, you have the opportunity to select one or more of these methods. For our More_Operations example, we select:
       com/IBM FileNet/churnett/ciMore_Operations.class

     **Important**: To complete this step, the Jar file must be accessible from your local computer. At this point, you import only the class name and methods. For runtime, the Jar file must be located in the Application Engine server where the Component Manager runs. Use the Process Task Manager to specify the exact location of the Jar file and to start the component.

   - **Java Class**: Select the appropriate class from the drop-down list of the available classes. After you select the Java class, a list of available methods display in the Available Methods box. When you configure the component queue operations, you have the opportunity to select one or more of these methods. For our More_Operations example, we select:
     com/IBM FileNet/churnett/ciMore_Operations.class

     Figure 8-4 shows the Jar File Path and Java Class set up for our example.

---

**Figure 8-4  Configure Java component: Jar File Path and Java Class**
– For the JMS component, set information for the following fields and click **OK**:

  * Queue Connection Factory (use JNDI)
  * Queue Name (JNDI)

  Java Naming and Directory Interface (JNDI) is a standard extension to the Java platform. It provides Java applications with a unified interface to multiple naming and directory services in the enterprise.

6. Set the Adaptor Properties:

   – Concurrent Threads: Number of concurrent threads to execute for the component. For More_Operations, we use the default value 1.

   – Polling Rate: The amount of time in milliseconds between polls of the queue by this component to look for work. For More_Operations, we use the default value 1000.

   – Exception Submap: Exception handling submap used by the component. For More_Operations, we select the default, **Malfunction**.

   – Startup Mode: Automatic or manual startup. For automatic start, the adaptor starts when the Component Manager starts. For More_Operations, we select the default, **Automatic**.

7. Set the JAAS Credentials:

   – User Name and Password: User name and password used for identification and permissions on both the Process Engine and potentially any external systems that will be accessed. For More_Operations, we use the administrator user ID and password.

   – Configuration Context: The context you enter must be defined in the taskman.login.config file (found in …\IBM FileNet\Router). This is the JAAS login configuration file that the CE_Operations component uses by default. It contains various login context sections that components use for authentication. For More_Operations, we enter CEl1ogin.

See Figure 8-5, which shows these selections.
8. Click Finish.

8.2.2 Importing component queue operations

After you create a new component queue, you must import operations from the selected Java class or specify JMS events.

To do so, follow these steps:

1. From the Process Configuration Console, right-click the new component queue you just created and select Properties from the context menu.
2. Select the Operations tab and click the Import icon. See Figure 8-6.
3. Select either the Java methods or the JMS events to import, depending on the adaptor type:

   - For the Java component, select one or more methods from the Available Methods drop-down list and click OK. The methods you select will be converted into operations. The method name becomes the operation name, and the method parameters become the operation parameters. In the Process Designer, you can then specify the operation and its parameters when designing the component steps of your workflow.

   You can optionally rename parameters and enter descriptions for operations and parameters. These can be helpful to the workflow designer when specifying expressions for the parameters.

   **Important:** If the imported methods depend on other Jar files, then you should make them available to the Process Configuration Console through the Register Additional Classes interface. See Figure 8-2 on page 276.

   Otherwise, the Process Configuration Console will fail to load the class and thus does not display any methods.

   These Jar files should also be added to the Required Libraries of Component Manager as described in 8.2.3, “Configuring Component Manager” on page 283.
See Figure 8-7, which shows these selections.

![Select Methods to Import](image)

*Figure 8-7  Select methods from Java class to import for the Java component queue*

**Note:** You cannot import Java methods that use byte, char data types, or any Java objects other than Short, Integer, Long, Float, Double or Boolean. You can use methods that use VWAttachment and VWParticipant.

For JMS operations, enter the events for this component queue. For each event, you enter an event name, and then specify the parameters for that event. All parameters that you specify have read-only access. You can optionally enter a description for each event and for each parameter.

The events will be converted into operations. In the Process Designer, you specify the expressions for parameters in the component step of your workflow.

The JMS adaptor only supports outgoing data to a JMS queue.

**Important:** Component Queue operations cannot be deleted later. After you commit the component queue definition, and it is included in a transferred workflow, you cannot delete the operations.

In addition, you must commit your changes and restart Component Manager before the new queue operations or changes take effect. Otherwise, the changes will be lost when you stop or refresh the Component Manager.
4. After importing the operations, the Operations tab should look similar to Figure 8-8. Click **OK**.

Before you proceed, you might want to add a description for each operation so that later, when using them in a workflow design, it is easier to see which ones you should use. If you imported the component with an XML file, the description might already have been provided.

![Component Properties showing imported operations](image1.png)

**Figure 8-8** Component properties showing imported operations

5. Commit the changes by right-clicking the isolated region and selecting **Commit Changes** from the context menu. See Figure 8-9.

![Commit changes for the component queue definition](image2.png)

**Figure 8-9** Commit changes for the component queue definition
8.2.3 Configuring Component Manager

In order to use the component queue that you created, you must configure the Component Manager to use the component queue.

The Process Task Manager in the Application Engine enables you to configure, start, and stop a Component Manager and its associated components. You can run one or more Component Managers. Each Component Manager coordinates one or more components which are responsible for delivering events from the Process Engine to an external entity such as a Web service or a messaging system. In the Process Task Manager, the Component Managers folder displays all the Component Managers currently defined. Selecting a Component Manager displays its associated components.

To configure a Component Manager to use the new component queue that you created, perform the following steps:

1. Start the Process Task Manager from the Application Engine.
2. Stop the Component Manager where you will add the component queue. You cannot update its information while it is running.
3. In the General tab, append the Queues field to include the new component queue name. Remember to add a comma in between value and the new name. There should be no space between them. For our More_Operations example, we append the queue name, More_Operations, at the end. See Figure 8-10.

![Figure 8-10  Add new component queue using Component Manager](image)
4. In the Required Libraries tab, specify the location of the Jar file that contains the class associated with the component queue. The Process Task Manager appends the Jar file location to the class path when starting the Component Manager. You must add a required library entry for each component queue that you create. Click Add to add the path. See Figure 8-11.

**Tip:** To see what component queues are available, click **Show queue names** on the right side of the Queues field. It shows a list of the component queue names. Make sure you append the exact name at the end of the Queues field.

**Note:** The Jar file must exist on your Application Engine server.

For our example, we placed the More_Operations.jar file in the Application Engine, C:\P8Components directory. So, we add the following path in the Required Libraries:

C:\P8Components\More_Operations.jar
5. Restart the Component Manager by right-click it and select **Start** from the context menu to make the changes effective (see Figure 8-12).

The Component Managers node allows a user to start and stop each Component Manager separately, as well as the individual component that has been defined. Starting the Component Manager also starts any components that have been configured to start automatically via the Process Configuration Console.

![Figure 8-12 Start the Component Manager](image)

### 8.2.4 Using components in a workflow

To use the component operations, add a component step in your workflow using the Process Designer. A component step represents an activity in the workflow that is assigned to one or more operations in a component queue.

Follow these steps to add component operations in a workflow:

1. Open your workflow definition.
2. Add the Component step from the palette below to your workflow map.
3. Select the new Component step from the map.
4. From the right side panel, in the Operations section, click the **Add** icon.
5. Select the component from the drop-down box. For our example, we select **More_Operations**.
6. A list of operations available for this component appears on the Operations section. Select one or more operations from this list. For our case, we select **copy** and **move**.
7. Select the operation one at a time to set its parameters. When you select an operation, the associated parameters appear on the bottom of the panel with their names, data types, and access rights.
An icon next to the parameter type indicates what access right the parameter has. Possible options include:

- Read: Reads from a work item.
- Write: Writes to a data field in the work item.
- Read/Write: Passes a value from the work item to the operation and then optionally updates the value when the operation is over.

The Expression field contains a named expression for your use in specifying the value for each parameter.

You can also specify the incoming and outgoing routing properties for the component step, similar to other step types.

**Note:** For step-by-step instructions, we describe how to add a component step in our case study; refer to 6.5.1, “Setting up a component step” on page 169.

---

### 8.3 Understanding Web services

Web services provide universal inter-operability between heterogeneous applications using Web standards. The Process Engine uses Web services technology to enable interaction between IBM FileNet Business Process Manager processes, as well as between IBM FileNet Business Process Manager processes and other business processes and applications.

IBM FileNet Business Process Manager processes can use (consume) the existing Web services. In addition, the processes can be published as Web services, to provide services to (be consumed by) others. The published Web services can be discovered through a UDDI registry. Using Process Configuration Console, you can configure a limited UDDI registry list for an isolated region or allow users to browse for published Web services.

#### 8.3.1 Web services adaptor

Component Integrator handles the interaction between the Process Engine and Web services (see Figure 8-13). This communication is transparent to both the workflow administrator and users. The architecture supports both polling and event driven modes for the Component Manager. You can configure the Component Manager to automatically respond to new events as they occur or poll for work.
As shown in Figure 8-13, the outbound requests for Web services are handled through *WS-Adapter*, running as part of the Component Integrator. The inbound requests for Web services are handled by *WS-Listener*, running as part of the Component Integrator. The Process Engine has a *WSRequest queue* and a *WSPending table*, which are system-managed data structures used in processing the requests for Web services.

All communications are accomplished through XML. When using Web services, you have to set data field mapping. There is no requirement for you to understand XML. The Process Designer tool automatically parses the WSDL for input and output parameters. The data fields are updated with the values of the incoming parameters at runtime.

**Invoking Web services**

To invoke a Web service in your workflow, you can create an Invoke step in the workflow. The Invoke step calls the Invoke system function, which allows you to call any application component that has been exposed as a Web service. This enables your workflow to interact with external Web services.

The Invoke function is implemented as an operation on the WSRequest queue. When the system encounters the Invoke call, the work item is routed to the WSRequest queue. The WS-Adapter, running under the control of Component Integrator, queries the queue and processes the work item. The work item is placed in the WSPending Table waiting for coming responses from the Web service. WS-Listener listens to any inbound request.
Repeating and receiving Web services

You can publish a process as a Web service to be used by other processes and applications. To handle inbound requests for your published Web services, use the Receive and Reply steps in your workflow.

The Receive step in a workflow calls the Receive system function. The Receive function can be viewed as representing an entry point to the workflow process. Incoming messages are processed by WS-Listener, which is running as part of Component Integrator. When processing incoming messages, the WS-Listener interacts with both the WSRequest queue and the WSPending table. If you put a Receive step immediately after a Launch Step in a workflow, an appropriate incoming message received by the system can launch a workflow process.

The Reply system function provides the ability to pass back information to the application that made the original request. There might be multiple Reply steps in a workflow process for one Receive step. The relationship between Receive and Reply is usually one to many.

The Reply step in a workflow calls the Reply system function. Similar to the Invoke function, the Reply function is implemented as an operation on the WSRequest queue. When the Process Engine encounters a Reply system function in a workflow, the work item is routed to the WSRequest queue. The WS-Adapter, running under the Component Integrator, queries the queue and processes the work item.

Unlike the Invoke function, there is no response received back from the call.

8.4 Implementing a process to invoke a Web service

The main steps to invoke a Web service can be summarized as follows:

1. Configure the UDDI registry list.
2. Define the Web service information in workflow properties.
3. Use an Invoke step in a workflow.

Note: For the step-by-step instructions that we used for our case study (excluding the configuration of the UDDI registry list), refer to 6.9, “Creating and calling a Web service” on page 230. There, we show you how to create a workflow process as a Web service, call the Web service in a workflow process, and test the process with Web services.
8.4.1 Configuring the UDDI registry list

Use the Process Configuration Console to configure the UDDI registry list for an isolated region. This registry list includes entries for Web services. You can search through this list to find the Web service to use or publish a workflow as a Web service in this registry.

Select or clear the check box option to allow or deny access from the Process Designer to any arbitrary WSDL. This controls access to Web services.

To configure the UDDI registry list:
1. Launch the Process Configuration Console tool.
2. Connect the isolated region (right-click the isolated region and select Connect from the context menu). See Figure 8-14.
3. Right-click the isolated region and select Properties from the context menu.
4. In the Isolated Region Properties window, select the UDDI tab and then select the UDDI Registry List tab.
5. Click the Add icon to add an entry to the list. See Figure 8-15.
6. Enter information for the new UDDI Registry entry:
   - Name: The name that identifies this UDDI registry entry. It will be displayed in the Process Designer when using the Web services Explorer.
   - Inquiry URL: The URL for inquiries to the registry. To validate a URL, select its row in the table, and click the **Validate** icon. This test confirms that an Inquiry URL is valid (can connect).
   - Publish check box: Select this option to enable publishing services to the UDDI registry specified by the Publish URL. The Publish URL, Business Name, Business Description, Business key, and User fields are optionally used when publishing a service.
   - Enable Process Designer to enter WSDL links without browsing for Web services: Select or clear the check box option. When this option is enabled, you can type in an arbitrary WSDL URL in the workflow properties Partner Links in the Process Designer. When this option is not enabled, you are restricted to search for WSDLs in the list of UDDI registries specified for this isolated region.

7. Click **OK** to close the Isolated Region Properties window.

8. Commit the changes by right-clicking the isolated region and select **Commit Changes** from the context menu.
8.4.2 Defining Web services information in workflow properties

In the Process Designer, you must define the workflow properties for the Web services that you want to invoke.

If a workflow invokes or receives requests for Web services, specify the Web services on the Partner Links tab. If XML data fields and XML schemas are required for invoke, receive, or reply tasks, define them in the XML Data Fields tab and the XML Schemas tab. On the Web services General tab, you can specify a folder in your object store or library where any incoming attachments are stored.

To enable your workflow to invoke a Web service:
1. Start the Process Designer, open the workflow definition.
2. Open the Workflow Properties window.
3. Select the **Web services** tab and then select the **Partner Links** tab.
4. Select the **Invoke** check box and click the **UDDI Explorer** icon. See Figure 8-16.

![Figure 8-16 Set Invoke WSDL URL](image)
5. Select **UDDI registries** (see Figure 8-17). Open the desired registry in the left pane. This displays any UDDI registries previously defined using the Process Configuration Console. Locate the Web service you want to invoke in the UDDI registries. The Web service can be either in public or private UDDI registries, or in workflows in the current isolated region that contain a Receive step. You can find the WSDL in the UDDI registries.

You can perform a keyword search or simply browse through the configured registry entries. To perform a keyword search, select either **Business list** or **WSDL tModel list**, then enter the first few letters of the business name or WSDL tModel name and click **Execute** to display the matching items. If a large number of items are returned, you can narrow the search by entering the whole name, or by using the Case sensitive and Exact match options.

![Figure 8-17 Select a Web service from UDDI registries](image)

6. Select the Web services you want to invoke.

7. Click **OK** to extract the WSDL information to the Partner Links table.
### 8.4.3 Defining an Invoke step in a workflow

To define an Invoke step in a workflow map:

1. Open the workflow definition.
2. Place an Invoke step on the workflow map using one of the following methods:
   - Use a System step:
     1. Add a System step to the workflow map.
     2. Select the System step icon from the workflow map.
     3. From the right panel under the Functions section, select **Invoke**. Use the right-arrow key to add it to the Selected Functions field. See Figure 8-18.

![Figure 8-18](image-url)  
*Use the System step with the Invoke function in workflow map*
Use a special Invoke step:

i. Select **Web services Palette** from the View Palette drop-down box on the bottom of the Process Designer window.

ii. Add the Invoke step to the workflow map. See Figure 8-19.

![Figure 8-19 Use Invoke System step from Web services Palette](image)

3. Specify the general properties for the Invoke system function. If you use the System step, double-click **Invoke** from the Selected Functions field. If you use the special Invoke step, select the step from the map and select the **General** tab on the right. Enter the following information:

- **Partner Link**: Select the Partner Link for the Web service. The Partner Link list contains only entries already specified for Invoke Web services in the workflow properties.
- **Operation**: Select the appropriate operation in the Web service.
– Message Type: The Process Designer automatically selects the message type. Two message types are:

  • Parameters: This uses the workflow data fields and attachments for input and output. This option is available for simple parameters.
  
  • XML: This uses XML messages for input and output. This option is for more complex messages.

If required, you can override the Parameters selection and select XML instead, but not vice versa.

– Outgoing Parameters: If you select the Parameter message type, and the Web service that you invoke requires input, then the required fields are displayed under this field. In the Expression field, specify a previously defined data field or expression of the appropriate data type.

– Incoming Parameters: If you select the Parameter message type, and the Web service will return data, then the name and data type are displayed under this field. In the Field Name field, select a previously defined data field of the appropriate type to hold the returned data. See Figure 8-20.

![Figure 8-20   General tab for the Invoke step](image)

4. Select the Advanced tab. In this tab, you can optionally set the time out expression for the step and the time out map that would be executed if a time out does occur. In addition, you can select the Use reliable messaging option here.

5. Make the appropriate routing connections between this Invoke step and other steps and save the workflow definition.
8.5 Implementing a process as a Web service

Before you turn a workflow process over to a Web service, make sure that the workflow works correctly as it is.

The main steps to implement a workflow process as a Web service can be summarized as follows:

1. Define the Web service information in workflow properties.
2. Use the Receive/Reply steps in a workflow.
3. Finalize the Web service operations and transfer the workflow definition.
4. Publish the workflow process as a Web service.

The workflow administrator uses Process Configuration Console to publish the Web service to the UDDI Registry List for the isolated region. Then external consumers can locate the Web service in the registry and call it.

**Note:** For the step-by-step instructions we used for our case study, refer to 6.9, “Creating and calling a Web service” on page 230 in which we show you how to create a workflow process as a Web service, calling the Web service in a workflow process, and testing the process with Web services.

8.5.1 Defining Web services information in workflow properties

In the Process Designer, define the workflow properties for the Web service. Specify the Partner Link and port definition. The Partner Link information is used when defining the Receive and Reply steps in a workflow.

To do so:

1. Start the Process Designer and open the workflow definition.
2. Open the Workflow Properties window.
3. Select the **Data Fields** tab and enter the data fields that will be used to hold the reply messages.
4. Select the **Web services** tab and then select the **Partner Links** tab.
5. Select the **Receive/Reply** check box (see Figure 8-21). Enter the following information:
   - Name: Enter the name for the Web service on the left panel.
   - Process Port Type: Enter a name for the port.
   - Process Role
8.5.2 Using the Receive/Reply steps in a workflow

In order for a workflow process to act as a Web service, the workflow must contain at least one Receive step that accepts requests and optionally a Reply step that returns messages to the consumers. We highly recommend testing the workflow to make sure that it works before turning it over to a Web service.

To use a Receive step in a workflow:

1. In the Process Designer, start the existing workflow.

   **Important**: A workflow that is intended to be published as a Web service must not contain any blanks in the workflow name.

2. Open the workflow definition.
3. Place a Reply step on the workflow map using one of the following methods:
   - Use a System step:
     1. Add a System step to the workflow map.
     2. Select the System step icon from the workflow map.
iii. From the right panel under the Functions section, select **Receive**. Use the right-arrow key to add it to the Selected Functions field. See Figure 8-22.

![Figure 8-22 Use the System step that calls the Receive function](image)

- Use a special Receive step:
  i. Select **Web services Palette** from the View Palette drop-down box.
  ii. Add the Receive step to the workflow map. See Figure 8-23.
Figure 8-23 Use Receive step from Web services Palette

**Important:** The Receive step must be the first step after the Launch step in order to launch this workflow automatically in response to an invoke of the Web service.

3. Specify the general properties for the Receive function. If you use the System step, double-click **Receive** from the Selected Functions field. If you use the special Receive step, select the step from the map and select the General tab. Enter the following information:

- Partner Link: Select the Partner Link for the Web service. The Partner Link list contains only entries already specified for Invoke Web services in the workflow properties.
- Operation: Select the appropriate operation in the Web service.
- Message Type: The Process Designer automatically selects the message type. Two message types are:
  - Parameters: Uses the workflow data fields and attachments for input and output. This option is available for simple parameters.
• XML: Uses XML messages for input and output. This option is for more complex messages.

If required, you can override the Parameters selection and select XML instead, but not vice versa.

– Outgoing Parameters: Specify outgoing parameter information.

**Note:** Because this workflow automatically launches when the Web service is invoked, do not assign F_Originator to any step in the workflow. Otherwise, it would not be considered a valid user.

4. Select the **Advanced** tab. In this tab, you can optionally set the time out expression for the step and the time out map that would be executed if a time out does occur. In addition, you can select whether to authenticate user or not and restrict the users (who can invoke this Web service) to specific users or groups.

5. Make the appropriate routing connections between this Receive step and other steps.

6. Optionally, add a Reply System step if required, to reply to the Web service consumer with its information.

7. Save the workflow definition.

### 8.5.3 Finalizing Web services operations and transferring the workflow definition

After you finish defining your workflow as a Web service, you must finalize the Web service operations in the workflow and then transfer the workflow definition to the Process Engine. Once you transfer a finalized workflow to the Process Engine, you no longer can make changes in the input and output parameters of the Web service operations.

To finalize the Web service operations and transfer the workflow definition:

1. In the Process Designer, open the workflow properties of the workflow.

2. Select the **Web services** tab and then select the **General** tab.

3. Select the **Finalize existing web services operations** option. See Figure 8-24. The system prompts you with a message regarding the implications of this action.
4. Click **Close**.
5. Save the workflow.
6. Transfer the workflow by select **File → Transfer**.

The Partner Link of the finalized Web service does not identify a specific version of the workflow. Other workflow processes always invoke the latest version of the transferred workflow.

**Note:** After you transfer the workflow definition to the Process Engine with the finalized option enabled, you cannot disable this option in the workflow definition.

### 8.5.4 Publishing the workflow process as a Web service

If a workflow process is to be consumed (used) as a Web service, you must deploy the workflow process as a WSDL definition and publish it to a UDDI registry. You can publish the workflow to either your private UDDI registry or in a public UDDI registry.

Before you can publish a workflow to a UDDI registry, you must specify the appropriate UDDI registry in the isolated region properties in Configuration Console.

After creating the workflow definition and transferring it into the Process Engine, use the Process Configuration Console to publish the workflow to a UDDI registry.
To publish a workflow process as a Web service:

1. In the Process Configuration Console, select the isolated region that contains the workflow to be published. The workflow must contain the Receive system function, and it must already have been transferred to the Process Engine.

2. Right-click the isolated region and select **Publish to UDDI** from the context menu.

3. In the Publish to UDDI window, select the **Publish Workflow** tab.

4. Enter a workflow definition name in the Search Web services workflows box, or leave the field blank and search for all workflows in the current isolated region that contains a Receive system function.

5. Click the **Search** icon to display the list of workflow definitions. Select the workflow to publish.

6. Select a previously specified UDDI registry. The list contains only those registries from which a valid certificate can be obtained.

7. Click **Publish**.

If you are publishing a business entity for the first time, enter the business name and description, and provide a user name and password for the UDDI registry. Otherwise, you cannot change the business name or user name. The information that you enter here displays in the UDDI Registry List for the isolated region.
Planning and designing a BPM solution

In this chapter we cover the process of planning and designing a content centric IBM FileNet Business Process Manager (BPM) application at a high level. This includes gathering requirements, transforming them into a workflow process, validating, and finally producing a high level architecture design.

We cover the following topics:
- Planning a BPM solution
- Functional design
- Detailed design

**Note:** The discussion we present here is intended for prototype application. To implement a production system, much more has to be taken into consideration. Contact your IBM FileNet lab services professional or appropriate business partners for details.
9.1 Planning a BPM solution

Planning and designing an IBM FileNet Business Process Manager (BPM) application is an iterative process. In this section we cover a walkthrough of the planning process at a high level. The intention is not to go deep to make it a step by step process. Rather, we want to provide you with an overall understanding of information gathering techniques, and validating and transforming requirements into a solution to a level where BPM process administrators would be able to use it for designing a prototype model in Process Designer.

The core discipline of business process management is a continuous process improvement. As the models that drive work or business transactions through the company evolve constantly, there is a need to have an agile method to model the process that IBM FileNet Business Process Manager offers.

Figure 9-1 shows the steps in modeling a process in an IBM Business Process Manager system.

![Figure 9-1 steps in modeling a process in a BPM system](image.png)
The inner circle of Figure 9-1 defines high level steps, and the outer circle outlines the detailed steps or the resulting actions to achieve the high level steps:

- **Discover:**
  - Action: Review
  - Action: Understand
  - Action: Evaluate

- **Design:**
  - Action: Process flow
  - Action: Integration
  - Action: Metrics
  - Action: Controls

- **Deploy:**
  - Action: User Interface

- **Monitor:**
  - Action: Analyze
  - Action: Optimize
  - Action: Simulate

Most outer circle actions require multiple iterations. Requirements gathering, designing, building, and testing all require multiple iterations to create and fine tune a process. The analyzing and optimizing phases involve several additional iterations as business analysts and managers experiment with alternations.

There are five major areas in the outer circle to look for when designing a BPM process: process flow, integration, user interface, roles and responsibilities, and controls.

**Process flow**
This action entails creating process definitions and flow. In the initial iteration, focus on the core functionality that delivers the bulk of the value.

**Integration**
This action focuses on integrating with and extracting information from internal and external systems, getting the required results, and updating databases.

**User interface**
After you understand the roles and responsibilities of all users, the next action is to ensure that the user interface windows deliver the required information and provide the right interface for users to interact with the system to perform their tasks.
Roles and responsibility
This action focuses on exploring the management information that is deemed necessary. Helpful questions to ask include: How the data is gathered, who should have access to the data, and how the data is presented.

Controls
Business managers want ways to improve performance of the process, allowing them to better control process execution. They require mechanisms that help them cater to peak system performance on demand, or influence the way in which business rules apply. This action focuses on providing the controls to the business managers.

The reiterated steps in modeling a business process ensures that no surprises emerge along the way, while delivering the flexibility to change as needed.

9.1.1 Discovering and understanding the process

Every organization has a different starting point with different requirements. Some organizations already have a defined process, while others are not as well developed. Some want to emphasize automation of the process, whereas others require better traceability, visibility, and performance measurement. Either way, the first objective is to understand the process. There are at least four possibilities for a business process:

- It is a brand new process.
- It is an ad-hoc workflow process.
- It is currently a paper-based manual process.
- It is an existing IBM FileNet Image Services (IS) eProcess.

The complexity of the discovery of a process varies based on the existing environment. You have to capture the process as it is. Assessing the existing process by using a set of complementary modeling techniques allows you to understand the process better. Ideal techniques for this phase include:

- Flow diagrams or flow charts to look at the order of activities
- Role activity diagrams, also called business context diagram, to focus attention on role interactions and the desired behavior of various actors

The emphasis here is on understanding the process, not building models for transformation into code or executable process definitions. This enables both the business analyst and the business users to step outside of the business-as-usual view and see the existing process differently.
The process of gathering required information about a business problem is termed a *walkthrough*. Before you start this process, we recommend that you prepare yourself and your audience for it.

### 9.1.2 Preparing for a walkthrough

As a good practice, we recommend preparing the audience for an interview or a walkthrough.

Before starting the requirement gathering process, conduct the following preparation steps:

- Prepare a list of topics and questions.
- Prepare the participants.
- Provide information about the process and products.
- Provide questions that require advance research.
- Plan for the expected amount of time and required degree of participation.

After you and your audience are ready, look for information in the following specific areas of the business:

- Organization and business objectives.
- Industry and comparable initiatives.
- Within current processes:
  - Departmental, inter-departmental, and external flows
  - Roles
  - Inefficiencies and associated costs
- Perceived gaps to desired business processes:
  - Functional requirements
  - Non-functional requirements
- Opportunities for improvement that have not been identified, such as:
  - Application integration
  - External process participants
  - Interactive process monitoring

### 9.1.3 Requirements gathering

In this phase, start gathering information by asking the related questions. The questioning technique differs in a great deal with the type and source of information. There are four methods of information gathering:

- Job shadowing
Each of the information-gathering techniques work best in a particular scenario. Pick and choose the technique that matches the target audience and the desired results. You might use a combination of techniques. Typically, use job shadowing and interviews with heads-down knowledge workers; use focus groups with supervisors and managers; and use questionnaires with occasional or external users.

**Job shadowing**
The job shadowing is primarily use for observing the current business process of workers below the supervisor level and observing corporate culture.

You can use job shadowing when the as-is state is important input to the requirements. Sometimes the requirements are better observed than described. Another reason for job shadowing is that the participants might have insufficient skills to communicate their requirements or current process, or are unwilling to communicate such information completely, or are intimidated in an interview environment.

While performing job shadowing, encourage demonstration of all business scenarios. Pay attention to the processes that are time sensitive, for example, specific to a certain time of the day, week, month, or quarter. Observe the manual handling of paper or electronic files. Check out the multiple ways and user interfaces used for the existing process. Also ask for spreadsheets, e-mail, paper logs, and phone calls that are part of the process.

**Interviews**
Use one-on-one interviews when a few individuals have specific unique knowledge or the requirements are better described than demonstrated, for example, when job shadowing is not appropriate.

You can use an interview primarily for management and executives, and for understanding high-level business objectives and long-term vision, the big picture.

**Focus group**
Use focus groups when knowledge is distributed over a number of individuals. The interaction between the participants is relevant to determine the requirements. The participants are peers (or nearly so) to reduce power-based domination.
You can use a focus group primarily for supervisors and senior workers who work closely and cooperatively on a daily basis. You can also use a focus group to reach consensus on shared requirements.

**Questionnaires**

Use questionnaires when there are many participants to survey or they are geographically remote. You can also use questionnaires for external constituents who might be inappropriate for internal meetings. Answers to specific well-defined questions, such as how many and how often, can also be surveyed in the form of questionnaires.

### 9.1.4 Interviews and questionnaires

When creating a list of questions, ask about how the current business processes are working and where they should be performed.

For managers and knowledge workers, it is often easiest to start with questions about the department, such as:

- What are the existing processes, including any process diagrams or procedures that they have created?
- How do they interface with other departments?
- How do they interface with external third parties systems, services, and customers?
- What are the existing and upcoming regulatory and customer service standards that they must meet?

As interview and questionnaires progress, you can move on to more specifics about their work, for example:

- What are the systems that they currently work with each day?
- What are the things that they find most difficult about the current processes, and what do they think can be changed to make it better?
- Are there any new reporting requirements that need to be implemented?
- What are the main software applications that they use to complete their work?
- What other applications do they make use of throughout the course of the day?
- Do they switch between applications frequently? If so, why? Do they re-key information from one system to another? If so, what is the information and why?
The end result of this process is to eventually compute a return on investment (ROI). The more information you gathered on both as-is and to-be states of the business processes, the easier it is to create a solid ROI statement. Whatever you ask, always determine the areas of current inefficiencies and the associated costs so the new or improved process can be better.

### 9.1.5 XYZ Corporation as-is auto claim approval process

By using the foregoing information gathering techniques, we have been able to determine the following facts about our case study of the XYZ Corporation auto claim process.

XYZ Corporation offers a range of financial and insurance products and services through a large network of field offices.

The key corporate business drivers include:

- Streamline™ business process efficiencies.
- Prove and maintain adherence to government regulations.
- Improve customer satisfaction.
- Increase competitive advantage.

Strategic objectives for the corporation include:

- Improve business processing.
- Grow business with less staff and leverage existing systems.
- Reduce cycle time for claims processing.
- Improve customer service.
- Launch self-service customer projects.
- Improve accounting.
- Better risk management.
- Control and manage e-mails.

From Information Technology (IT) perspectives, objectives include:

- Establishing new standards.
- Reducing IT expenditure through standardization and consolidation.

After gathering the existing process requirements and business initiatives of XYZ Corporation, we create a flowchart describing the current as-is auto claim approval process as shown in Figure 9-2.
Figure 9-2  Flowchart describing XYZ Corporation as-is auto claim approval process
After reviewing the process, we discover the following pain points associated with the current as-is auto claim process:

- It is manual and time intensive.
- There is a lot of rework due to errors.
- Many hours of overtime are required to correct errors.
- Documents are often not signed as required.
- Claimants have to wait a long time to receive checks.
- The file system is manual; not only is it time consuming to use, but also difficult to find the information in it later.

9.2 Functional design

In this section, we discuss analyzing the requirements and transforming the requirements into an IBM FileNet Business Process Manager functional design. We show that using IBM FileNet Business Process Manager improves the overall process from both tangible and intangible perspectives.

9.2.1 Transforming requirements into a solution

After you have gathered detailed requirements, identify and analyze the key areas where IBM FileNet Business Process Manager can be used to provide greater benefits.

Identifying the key requirements

First, make sure that the business process is important for business, and that the improvement of this process is critical to long-term success of the company or organization. Next, ensure that there is an easily identifiable role for IBM FileNet Business Process Manager as part of the process improvement.

Analyzing the requirements

Analyzing the requirements determine how a manual process can be redesigned to improve processing time. As a result, some part of the process can be automated and some part might be completely eliminated.

In our case study of auto claim process, we can eliminate the mail-room operation. This alone saves labor hours thus improving processing time and reducing the claim processing errors and overall costs.

We also look to automate certain areas for improvements. This includes identifying and assigning field agents and adjustors based upon accident location, type of a claim, and the claim amount involved.
Identifying a key business process

Identify a key business process that can be improved through IBM FileNet Business Process Manager.

Map the current as-is and future (or want) to-be business processes. Analyze the future business processes and find out more information about these business processes. This includes identifying the following information:

- The functions and roles within the process
- The source of content and its method of acquisition within the process
- Steps that can be automated
- Integration point within the process

Overlay the IBM FileNet Business Process Manager to check for the right fit, and identify the potential gaps in functionality between the as-is and to-be business process.

The graphical view of the current requirements is presented in a flowchart in Figure 9-3 on page 314. We also create a business context diagram, Figure 9-4 on page 316, to show user roles and interaction within the system.

9.2.2 XYZ Corporation future (to-be) auto claim approval process

In 9.1.5, “XYZ Corporation as-is auto claim approval process” on page 310, we discuss XYZ Corporation’s as-is process with a flowchart. In this section, based on our analysis of the process and XYZ Corporation requirements, we transform the as-is process into a future to-be process shown in Figure 9-3.

The purpose of having a future to-be process is to ensure that we understand the process in its entirety, including the roles and responsibilities of each player in the process. This way, we can get an agreement on the process steps and on the more low level requirements.
For auto claims, the comparison of the as-is process and the to-be process clearly shows a considerable reduction from 30 steps to 13 steps. This is done by eliminating mail-room operation and automating manual processes such as field agent and adjustor assignments.
The auto claim process depicted in Figure 9-3 on page 314 has four roles:

- **Customer Service Representative (CSR)**
  
  This is an individual, customer facing role. CSRs take customers’ calls and fill out eForms with claim information such as customer name, policy number, type of incident, and location of incident. After a CSR submits an eForm, a process is automatically initiated in the IBM FileNet Business Process Manager system.

- **Field agent**
  
  After the system launches a claim process, it finds the appropriate field agent and adjustor depending upon the location of accident, the type of claim, and the value of claim. The field agent then completes any missing information, including gathering supported documents such as police report and claim estimates, and routes the claim to an adjustor.

- **Adjustor**
  
  The adjustor receives the claim package from the inbox, reviews the claim, checks and validates the policy document, and takes one of three actions: approve the claim, reject the claim, or escalate it to supervisor due to certain exceptions. If the adjustor approves the claim, the claim goes for payment and an approval letter is sent to the claimant. If the adjustor rejects the claim, a rejection letter is sent to the claimant. In the case of escalation, the claim along with all supporting documents moves to the supervisor’s inbox for further processing.

- **Supervisor**
  
  Supervisor performs thoroughly reviews the claimant folder for supporting documents, any previous claims. If required, the supervisor runs a fraud investigation and risk assessment.

After identifying the business process and the responsible roles, we create a business context diagram that explains users’ roles, connectivity, and information flow for the auto claim process. See Figure 9-4.
Introducing IBM FileNet Business Process Manager

Figure 9-4  Business context diagram with roles and connectivity for auto claim process

Using both the to-be process and the business context diagram, you can clarify your understanding of the process and gain your customer’s commitment on the requirements.
9.2.3 Locking down the requirements

Locking down the requirements serves two purposes. The first purpose is to gain agreement from customers, including business users, IT staff, and the management. The second purpose is to have a base process to start creating a prototype solution. The best practice is to have a formal review with all partners to achieve a common understanding and consensus.

Address the following points when locking down the requirements:

- Base requirements
- Associated costs
- Process improvement requirements
- Discussion of the solution scenario
- Discussion of the stages of implementation
- High level overview of IBM FileNet Business Process Manager products that address the requirements
- Benefit summary

9.3 Detailed design

There might be gaps between the default functions provided by the IBM FileNet Business Process Manager system as-is and the customer requirements. Identify the alternatives. If the requirements are nearly all met, and the gaps are not significant, you might need minor customization. If the gaps are significant, either a third-party solution or add-on products can be used to close the gaps.

The alternatives will become part of the detailed solution design, wherein you describe the readily available functions as-is and any of the added functions from customization.

For the purpose of our case study, we use two interfaces. The eForms interface is used by CSRs who take customers’ calls and prepare claims. The eForms helps in filling the required information interactively while the customer is on the phone. After the process is initiated, a case is created. Other users, such as field agent, adjustor, and supervisor, require a review of the case. Therefore they use a customized user interface that provides more functions, such as reviewing the claimant folder, and associating more documents with the case, such as police reports, creating comments, and approving or rejecting the claims.
9.3.1 Solution components

After you have identified the key requirements and the business process, the next step is to list the requirements and map it with IBM FileNet Business Process Manager features in order to identify any gaps.

In our case study, we map the key business requirements with the features provided by IBM FileNet Business Process Manager and other IBM FileNet P8 products. The result is shown in Table 9-1.

Note: For the actual implementation of the case study in Chapter 6, “Implementing business processes: Case study” on page 127, in order to simplify the explanation, we do not show an eForms implementation. We use Workplace (XT) and the generic step processor provided with IBM FileNet Business Process Manager to complete the entire implementation.

<table>
<thead>
<tr>
<th>Business requirements</th>
<th>IBM FileNet P8 features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replace manual steps of creating the claim, such as creating and faxing word documents, time-stamping, classifying, and reviewing for input errors for the claim.</td>
<td>Use eForms to fully automate this process.</td>
</tr>
<tr>
<td>Replace the manual process of locating the policy document from a manual filing system, printing, and associating the policy with the claim package and re-filing.</td>
<td>Through the BPF customization, a case is created automatically after the claim is submitted. The claim and all associated documents are part of the case folder.</td>
</tr>
<tr>
<td>Replace manual interaction between field agent and adjustor, which has usually delayed the claim processing.</td>
<td>Interaction through the IBM FileNet Business Process Manager system, with built-in Time Delay, Exception, and Escalation triggers.</td>
</tr>
<tr>
<td>Replace manual handling of associated documents that were causing delay in the processing, for example, police reports, car photos, and so on.</td>
<td>The Content Engine used within IBM FileNet Business Process Manager stores these documents electronically.</td>
</tr>
<tr>
<td>Improve risk assessment, which currently is completely manual, using the previous claims history.</td>
<td>Intelligence is built into the IBM FileNet P8 process. Based upon the risk assessment rules, the adjustor can make informed decisions.</td>
</tr>
</tbody>
</table>
We identified a number of manual processes that can be fully automated with readily available IBM FileNet Business Process Manager functions. We identified at least three major requirements such as case management, push method of delivering the case to users, and comprehensive audit.

As we learned earlier, the business and strategic objectives of XYZ Corporation are to gain a competitive edge, so we need to have a solution that can be deployed in a short period of time. To address this and the other key requirements such as case management, comprehensive audit, and using a push method to deliver a case to adjustor’s queue, our plan is to use IBM FileNet Business Process Framework (BPF) in conjunction with IBM FileNet Business Process Manager.

While the BPF user interface is critical to claim processors, it is “overkill” for CSR's role. We decide to use a simple eForms-based user interface. A form can be used as a document or can be customized as an intelligent user interface. For our case study, we use the document type eForms.

<table>
<thead>
<tr>
<th>Business requirements</th>
<th>IBM FileNet P8 features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce claim approval’s overall processing time.</td>
<td>IBM FileNet Business Process Manager’s process has built in logic for claim approval, rejection, and escalation decisions based upon claim amount and corporate rules.</td>
</tr>
<tr>
<td>Improve payment process, which takes up time, due to not having complete information.</td>
<td>The claim case would have every document that is needed for a final payment.</td>
</tr>
<tr>
<td>Replace or improve the writing and dispatching of letters to the claimant, which currently are manual processes.</td>
<td>This is completely automated in the process step.</td>
</tr>
<tr>
<td>Provide status about the process, including who did what, which currently is hard to track down.</td>
<td>Comprehensive auditing in BPF Business Activity Monitor.</td>
</tr>
<tr>
<td>Properly retain and destroy claim files and associated documents, such as e-mails, according to corporate compliance policies, a feature that currently the manual process does not offer.</td>
<td>IBM FileNet P8 compliance framework can be proposed to address this requirement.</td>
</tr>
</tbody>
</table>
By automating the process steps in the IBM FileNet Business Process Manager system, we address a number of XYZ Corporation business objectives along with the strategic and the IT objectives such as improving customer satisfaction by reducing overall process cycle time. Automating the process also reduces labor time and errors, so more claims can now be handled correctly and speedily by the same number of people, thus reducing the overall cost per claim.

Besides automating the claim process, XYZ Corporation also ask for monitoring the claim process and using the results for further process improvements. This can be addressed by implementing Process Analyzer, Process Simulator, and Business Activity Monitor BAM.

As XYZ Corporation is looking to implement IBM FileNet Business Process Manager as an enterprise solution, they prefer to ensure that the system can scale, expand, and be able to handle future growth.

Compliance is another critical requirement that includes storing, discovering, and retaining claim files, e-mails, and other related documents for a specific period of time, and being able to disposed them according to corporate governance and regulatory requirements.

IBM FileNet Business Process Manager can use various high availability options, including server farms for scalability and expendability; while IBM FileNet Records Manager, Email Manager, and Record Crawler can be used in conjunction to address the compliance requirements.

### 9.3.2 Modeling the process

In this phase, we design a high level solution that shows process steps, roles, the user’s interactions with respective user interfaces, and process logic. All the design is based around IBM FileNet Business Process Manager and the additional components, in this case, eForms and BPF.

Finally we model the process in Visio using business process modeling notation, BPMN. The final BPMN design is shown in Figure 9-5. It can be imported as a BPM process design diagram into Process Designer (see Figure 9-7 on page 322).
9.3.3 Technical architecture overview

The system context diagram shown in Figure 9-5 provides a view of the proposed solution. For IT infrastructure, we use an architectural overview diagram so that XYZ Corporation IT staff can see both the logical and physical fit within their IT infrastructure.

Technical architecture is designed based upon the analysis of the requirements and identifying a process that shows a prototype solution. Some key reasons for creating a technical architecture are:

- To show how the solution fits within the existing IT infrastructure
- To map the solution features with the required functions
- To show connectivity to the internal and external systems
- To evaluate the delta between required and proposed specifications and its impact on the development approach
- To use it as a prototype to present solution’s functionality

Figure 9-6 shows the created technical architecture of our auto claim process. The auto claim BPM process design diagram is shown in Figure 9-7.
Introducing IBM FileNet Business Process Manager

Figure 9-6  High level architecture for XYZ Corporation prototype auto claim process

Figure 9-7  Auto claim BPM process design diagram
9.3.4 Cost analysis metrics

As we discussed earlier, business process management relates to change management that results in improvements in several business areas. Some of these areas include:

- Elimination of manual labor, such as a process step that automatically sends an email transaction confirmation to a customer, which in turn replaces a manual process.

  For the auto claim process, we identified the following manual processes:
  - Manual filing and re-filing
  - Mail room handling
  - Calls/email

- Capturing of process information for understanding, such as the automated gathering of data showing when each item in a process is entered and exited at each processing step.

- Improvement in analysis of information and decision-making, for example, based on current data, calculate the average wait time for an item in each process step. In the current auto claim process, it would be several work hours.

- Coordination of processes across distances, making processes independent of geography, for example, a data entry process step executed by operators in home-based or remote offices, regardless of where the process initiated. In other words, CSRs and adjustors are location independent.

- Coordination between tasks and processes, such as a process step that automatically looks up client information in a line of business system based on a client account number in a CRM system that triggered the process. CSRs’ call time are greatly reduced.

- Capturing, storing, searching, and distributing of unstructured content by providing direct online access to content such as claim cases and associated documents in IBM FileNet P8 content repository.

There are two areas that directly impact ROI metrics:

- Cost reduction metrics
  This is typically supportable with hard data and is generally well-understood by all levels.

- Competitive advantage metrics
  This requires revenue forecasting but might have a supportable model. It might be understood only at executive and management levels.
The cost reduction metrics provide tangible benefits in the following areas:

- Reduced error rates
- Increased process efficiency
- Reduced/consistent end-to-end service time (cost of compliance failure)
- Reduced time to implement process changes
- Increased process visibility (cost savings through self service)
- Reduced effort to measure process
- For document-driven processes, reduced manual distribution

The competitive metrics provide intangible benefits in the following areas:

- Improved time to market due to process improvement and greater agility
- Reduced/consistent end-to-end service time
- Increased process visibility (for example, the quality of customer service)
- Increased capacity
- Technology as a differentiated against competition
- Increased efficiency from handling customer inquiries on-line

9.3.5 Final walkthrough

In the final walkthrough, our objective is to gain agreements on what has been done so far and proceed to the next step of creating a prototype solution, followed by a customized demonstration and a high level return on investment (ROI) for both hard and soft dollar savings.

The walkthrough discusses:

- Detailed solution design
- Configuration and process design
- Customizing and integration with other systems
- Monitoring and reporting
- Process management
- Customized product demonstration; using simulation and analysis, an ROI can be demonstrated
- Return on investment on a business case

Implementing IBM FileNet Business Process Manager means introducing change management. It is important to talk about it and point to any additional savings due to changes in the process.
Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

ftp://www.redbooks.ibm.com/redbooks/SG247509

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the Additional materials and open the directory that corresponds with the IBM Redbooks form number, SG24-7509.
Using the Web material

The additional Web material that accompanies this book includes the following file:

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG247509.zip</td>
<td>Contains object store data model and workflow definition used for the case study.</td>
</tr>
</tbody>
</table>

System requirements for downloading the Web material

The following system configuration is recommended:

- **Hard disk space**: 40 MB minimum
- **Operating System**: Windows XP
- **Processor**: Pentinum IV
- **Memory**: 512 MB

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder. Follow the instruction in the README.txt file.
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 327. Note that some of the documents referenced here might be available in softcopy only.

- *IBM FileNet Content Manager Implementation Best Practices and Recommendations*, SG24-7547

Online resources

These Web sites are also relevant as further information sources:

- Product documentation for the IBM FileNet P8 Platform
  

  This is where you can download the IBM FileNet P8 Platform ecm_help documentation (ecm_help.zip), FileNet P8 System Overview (in PDF), and many other valuable product documentation.

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)
Help from IBM

IBM Support and downloads
ibm.com/support

IBM Global Services
ibm.com/services
Index

A
abdication 61
active content 4
Adjustor Review 148, 151
APIs
  Process Engine 10, 12, 43
application
  IBM FileNet Enterprise Manager 129
  Process Administrator 43, 205–206, 210–211
  Process Analyzer 6, 13–16, 44, 320
  Process Configuration Console 43, 268–269, 274
  Process Designer 5–6, 9, 39, 43, 47, 128, 132, 158, 267, 269, 274, 276, 280–281, 304, 320
  Process Simulator 5–6, 13–14, 42, 44, 320
  Process Task Manager 43
  Step Processor 43
  Task Manager 283
Tracker 43, 150, 205–206
tracker 5
Application Engine (AE) 13, 36, 39
attachment 57, 139
attachments
  workflow definition 82
automation
  process 2

B
BAM Dashboard 46
Base Workflow 78–79
BPF_Operations 260
BPM Palette 72, 90, 146–147
Business Activity Monitor 42
Business Activity Monitor (BAM) 13–14, 16, 45, 320
definition 2
  escalation mechanism 60
first step 110
key elements 55
Web services 49
Business Process Framework (BPF) 11, 46
business process management
  active content 4
definition 2
  primary focus 2
  transactional and content centric processes 3
business requirement 136
business rule 28, 34, 267, 269, 306
business rules engines 13

C
case 11
case management model 11
case study 127, 129, 132, 135–136, 310, 312
  claim approval process 24, 26–28
CE_Operations 258–259
claim approval process 24, 26–28, 56, 310, 313
claim folder 169, 172–173, 176
claim process 212, 227, 311
Claim Setup submap
  step 177, 182
collector step 59, 99, 104
component adaptor 272
Component Integrator 5, 8, 12, 39, 43, 65, 128, 258–259, 271–272
  runtime interaction 272
  use case examples 259
component integrator 12
Component Manager 272, 274, 277
component queue 274
  importing operations 279
  required library entry 284
component queues 272, 276
candidate step
  incoming and outgoing routing properties 286
conditional route 114, 128, 176
conditional routing 59
content
  active content 4
definition 3
  federated 19
content centric process 3
Content Engine (CE) 13, 21, 36, 44, 129, 258

© Copyright IBM Corp. 2008. All rights reserved.
Introducing IBM FileNet Business Process Manager

database 42
federates content 19
operations 259
workflow process 143
workflow process definition 142
Content Engine (CE) Kernel 41
Content Federation Services for Image Services (CFS-IS) 19
Content Manager 19

D
data field 59, 81, 136–137, 267, 286–287
data tier 42
data type 11, 81, 136, 281, 285
database lookups 28
deadline 60
  definition 9
delegation 61
Directory Services 42

E
eForms 10–11, 27
electronic form 27, 31, 38
Email Manager 21
e-mail notification
disable 81
engine
  definition 36
enterprise content management (ECM) 10, 18
Enterprise Information System (EIS) 258
escalation
  business process 60
Event Dispatcher 44
event log 44, 49, 51, 80, 117, 136
  Process Engine 15
external system 29–30

F
F_Trackers 57
farm
  servers 50
federated content 19
FileNet System Monitor (FSM) 21
focus group 308

G
General step 8, 72–73, 91, 160

I
IBM FileNet Business Process Framework (BPF) 10
IBM FileNet Content Manager 19
IBM FileNet Email Manager 21
IBM FileNet Enterprise Manager 129
IBM FileNet P8 Platform 36
IBM FileNet Records Crawler 21
IBM FileNet Records Manager 20
IBM FileNet System Manager 21
IBM FileNet System Monitor 21
IBM FileNet Workplace 10, 12
import
  XML file 275
incoming message 40, 288
inheritance
  workflow 117
Instruction Sheet Interpreter (ISI) 268
integration options
  process 5
interview 307–308
Invoke 264
Invoke step 293
invoke Web services 287
isolated region 49–50, 79, 275, 282, 286, 289
  limited UDDI registry list 286
  UDDI Registry List 296
  UDDI registry list 289, 302

J
Java Authentication and Authorization Service (JAAS) 273
Java class 272
Java Message Service (JMS) 39
Java Messaging System (JMS) 12
Java Naming and Directory Interface (JNDI) 278
JMS 258
job shadowing 308

K
Key Performance Indicators (KPIs) 16, 45

L
Launch step 8, 110, 161, 265, 288, 299
Lotus Forms 11

M
main workflow map 128, 184
manual process 2, 24, 26, 306, 312, 319
map
  definition 9
  malfunction map 121
  workflow map 70–71, 91, 128, 142, 184, 242,
    268, 285, 293
message 40
message type 261, 295
Microsoft Visio 5–6
middleware tier 41
modeling
  process 4
modeling process 304, 320
monitoring 5
  process 6
More_Operations 169, 172, 260, 275–276
multiple routes 150, 152

O
object store 19, 42, 82, 87, 138–139, 259, 291
  attachment item 139
  attachment template 139
  definition 36
On-Line Analytical Processing (OLAP) 15
operation parameter 170, 231
operations
  import, component queue 279
optimization 5
  process 3

P
palette 72
  BPM Palette 72, 90, 146–147
parameter 161
participant 55
Partner link 87, 122, 233, 290, 292, 294, 296, 299,
  301
  Web services 291
police report 30, 56, 130, 173, 315, 317
policy holder 24, 30, 55–56
portlet 10, 12
presentation tier 39
process
  active content 4
  business process definition 2
  case study 24, 26–28
  content centric 3
  deadline definition 9
  disable email notification 81
  engine definition 36
  integration options 5
  integration to external systems 3
  isolated region 79
  map definition 9
  map, malfunction map 121
  map, workflow map 70–71, 91, 128, 142, 184,
    242, 268, 285, 293
  modeling 304, 320
  palette 72, 146–147
  process voting 61
  queue, work queue 164
  route definition 8
  routing 58
  routing, conditional route 114, 128, 176
  routing, conditional routing 59
  routing, multiple routes 150, 152
  routing, simple majority 62
  routing, single routing 58
  step definition 7, 58
  step, collector step 59, 99, 104
  step, Component step 8, 72–73, 108, 160, 169,
    274, 280–281
  step, first step 110
  step, General step 8, 72–73, 91, 160
  step, Invoke step 293
  step, Launch step 8, 110, 161, 265, 288, 299
  step, Reply step 288, 296–297
  step, reply step 49, 231, 245
  step, Submap step 72, 84, 147, 160, 172, 185
  step, System step 72–73, 105, 181, 266, 293,
    297
subprocess 9
subscription 192, 202
system function 73, 80, 105
  Reply 87
task 55
timer definition 9
transactional 3
use case 56, 310, 313
Web services 255–256
work queue 51
workflow definition 68, 79–80
Introducing IBM FileNet Business Process Manager

workflow process 143
workflow process definition 142
workflow properties 71
XML process definition language (XPDL) 5–6
Process Administrator 43, 205–206, 210–211
Process Analyzer 6, 13–16, 41, 44, 320
database 42
OLAP 42
process automation 2
process configuration 39
Process Configuration Console 43, 268–269, 274
process definition 6, 9, 51, 305–306
definition 7
Process Designer 5–6, 9, 39, 43, 47, 128, 132, 158, 267, 269, 274, 276, 280–281, 304, 320
calendar queues 276
malfunction map 121
step palette 72
workflow properties 71
process designing 4
Process Engine (PE) 4, 6, 13, 15, 21, 36, 43, 47, 49–51, 129, 186, 258, 261, 272–273
APIs 10, 12, 43
calendar 36
components 47
database 42, 44, 46, 50
event log 15
logical view 49
server 50–51, 272, 275
system 41, 50
Process Engine (PE) Kernel 41
process flow 305
process improvement cycle 14
process instance
definition 51
process modeling 4
process monitoring 5–6
process optimization 3, 5
Process Orchestration 40
process orchestration 263–264
process server 50
Process Simulation Console 44
Process Simulation Designer 44
Process Simulator 5–6, 13–14, 42, 44, 320
process subscription 191
Process Task Manager 43, 283
Process Tracker 150, 205–206
process voting 61
properties 71
properties pane 71, 148, 152, 159
select condition 235
Publisher 44

Q
questionnaire 309
queue
component queue 274
component queue, import operations 279
definition 51
Instruction Sheet Interpreter 268
work queue 164
queues
component queues 272, 276

R
Receive system function 264–265
receive Web services 288
Records Crawler 21
Records Manager 20
Redbooks Web site 327
Contact us xiv
region
isolated region 49–50, 79, 275, 282, 286, 289
Reply function 87
Reply step 288, 296–297
reply step 49, 231, 245
Reply system function 264
reply Web services 288
requirements gathering 305
Return on investment (ROI) 310, 323
role 55
roster 80
definition 51
route 7
conditional route 114, 128, 176
definition 8
parallel 9
routes
multiple routes 150, 152
routing 28, 58
conditional routing 59
simple majority 62
single routing 58
Rules connectivity framework (RCF) 267
rules engine 41, 267–268
rules engine framework 5
Rules Framework 41
### S
- Sample solution: 55
- Search template: 138–139, 260
- Service level agreement (SLA): 16
- Service Oriented Architecture (SOA): 47
- Service Oriented Architecture: 48
- Simple majority: 62
- Single routing: 58
- Single-sign-on (SSO): 42
- Step: 7
  - Collector step: 59, 99, 104
  - Definition: 7, 58
  - First step: 110
  - General step: 8, 72–73, 91, 160
  - Invoke step: 293
  - Launch step: 8, 110, 161, 265, 288, 299
  - Reply step: 288, 296–297
  - Reply step: 49, 231, 245
  - Submap step: 8, 72, 84, 147, 160, 172, 185
  - System step: 8, 72–73, 105, 181, 266, 293, 297
- Step Processor: 43
- Submap step: 8, 72, 84, 147, 160, 172, 185
- Subprocess: 9
- Subscription: 191–192, 202
- System function: 73, 80, 181, 287–288, 294
  - Reply function: 87
- System functions
  - Invoke: 264
  - Receive: 264–265
  - Reply: 264
- System Manager: 21
- System Monitor: 21
- System property: 199, 201
  - Symbolic name: 202
- System queue: 51
- System step: 8, 72–73, 105, 181, 266, 293, 297

### T
- Target object
  - Class: 188–189, 198
  - Class type: 188
  - Information: 216
- Task: 55
- Task Manager: 43, 283

### U
- UDDI registry: 262, 264
- UDDI registry entry: 290
- UDDI registry list: 286, 289
- Universal Description, Discovery and Integration (UDDI): 261–262
- Use case: 55
  - Claim approval process: 56, 310, 313
- User interface: 164, 305, 319
- User queue: 51

### V
- Voting
  - Process voting: 61

### W
- WaitForCondition: 80
- Walkthrough: 307, 324
- Web Application Toolkit (WAT): 10, 12
  - Inbound requests: 287
  - Invoke: 287
  - Outbound requests: 287
Introducing IBM FileNet Business Process Manager
Introducing IBM FileNet Business Process Manager

This IBM Redbooks publication provides a basic introduction to IBM FileNet Business Process Manager (BPM) V4.0. BPM enables organizations to create, modify, and manage content centric business processes. One key advantage of BPM is its ability to work with active content, which refers to the ability of content to trigger or affect business processes.

In this book, we cover the key elements that make up a business process, including tasks, participants, roles, steps, routing, and deadlines. We describe how to use Process Designer (a BPM application) to design your business processes. In addition, we provide step-by-step instructions on how to implement a use case business process scenario.

BPM also supports integration with external systems and services through Component Integrator, Web services, and Rules Connectivity Framework. We discuss these integration options and provide instructions on how to implement the use case scenario by demonstrating some of the options.

This book is useful for system architects, process analysts, and process designers who require an understanding of IBM FileNet Business Process Manager. It also serves as a practical guide for those who want detailed instructions in order to implement a BPM system.

For more information: ibm.com/redbooks