

Governing Operational Decisions in an Enterprise Scalable Way

Jonathon Carr

Eric Charpentier

Nigel Crowther

Guillaume Devillers

Sophie Fargier

Julie Holm



 **Cloud**



International Technical Support Organization

**Governing Operational Decisions in an Enterprise
Scalable Way**

July 2017

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

First Edition (July 2017)

This edition applies to Version 8.9.0.1 of IBM Operational Decision Manager.

© Copyright International Business Machines Corporation 2017. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
Preface	ix
Authors	ix
Now you can become a published author, too	xii
Comments welcome	xii
Stay connected to IBM Redbooks	xii
Chapter 1. Introduction	1
1.1 Assumptions	3
1.1.1 The ODM components	3
1.2 Decision governance	4
1.2.1 The Benefits of decision governance	4
1.2.2 The Evolution of decision governance	4
1.3 The sample scenario	5
1.3.1 Loan Validation background	5
1.3.2 The Loan Validation solution	6
1.4 Book layout	6
Chapter 2. Decision governance for project managers	9
2.1 Staffing the project team	10
2.2 Planning for decision governance	10
2.2.1 Governance within the first iteration	10
2.2.2 Governance tasks	12
2.2.3 Governance only process	14
2.3 Project risks and problems associated with decision governance	15
2.4 Changing your organization to realize your decision governance vision	16
2.4.1 The role of the Center of Excellence	18
2.5 Conclusion	19
Chapter 3. Roles and responsibilities in governing decisions	21
3.1 Decision governance roles	22
3.2 Functional roles and responsibilities	23
3.2.1 Business roles	23
3.2.2 Business users	24
3.2.3 Technical roles	25
3.3 Secondary roles and responsibilities	26
3.3.1 Secondary business roles	26
3.3.2 Secondary technical roles	27
3.4 Clarifications about roles	28
3.5 Conclusion	29
Chapter 4. Securing the Decision Center	31
4.1 Authentication	32
4.2 Access control levels	32
4.3 Permission patterns	34
4.4 Security configuration process	35
4.5 Sample team	35

4.6 Simple pattern	36
4.6.1 The Predefined Decision Center UI roles	37
4.6.2 Configuring WebSphere Application Server users and groups.	38
4.6.3 Configuring Decision Center groups to roles mapping	39
4.7 Fine-grained pattern	40
4.7.1 Decision Center group permissions	40
4.7.2 Decision Center roles permission requirements	41
4.7.3 Decision Center Permissions Groups	42
4.7.4 Creating Decision Center users and groups	44
4.7.5 Enforcing Decision Center project security	46
4.7.6 Configuring Decision Center permissions	47
4.8 Project pattern	50
4.8.1 Defining LDAP Users & Groups	53
4.8.2 Adding LDAP registry to WebSphere Application Server Global Security	53
4.8.3 Configuring Decision Center role-to-group mapping	53
4.8.4 Configuring a Decision Center LDAP connection	54
4.8.5 Uploading LDAP Users and Groups in Decision Center	55
4.8.6 Configuring Decision Center project security	57
4.8.7 Configuring Decision Center permissions	58
4.9 Conclusion	59
Chapter 5. Designing decision services	61
5.1 Concepts	62
5.1.1 Decision service	62
5.1.2 Execution object model	64
5.1.3 Business object model	64
5.2 Project hierarchies	64
5.2.1 Diamond-shaped project hierarchy	65
5.2.2 Our sample project hierarchy evolution.	66
5.3 Folder structure	68
5.4 Orchestrating ruleset execution	69
5.5 Advanced topics	70
5.5.1 Adding projects	70
5.6 Conclusion	71
Chapter 6. Processes	73
6.1 Introducing the business release process	74
6.1.1 Change activities	74
6.1.2 Validation activities	75
6.2 Documenting processes	75
6.3 Sample processes	75
6.4 The IT Centric Process	76
6.4.1 The initial IT release	76
6.4.2 Staying in IT: IT release for subsequent releases	77
6.5 The business release	78
6.5.1 Simple business rule change	79
6.5.2 Complex business rule change process	80
6.6 The technical release	80
6.7 Conclusion	83
Chapter 7. Decision governance framework	85
7.1 Decision governance framework lifecycle	86
7.1.1 Release lifecycle	86
7.1.2 Change activity lifecycle	87

7.1.3 Validation activity lifecycle.	88
7.2 A decision governance framework workflow pattern	89
7.2.1 The initial release	89
7.2.2 The first IT release	90
7.2.3 Validation activities	91
7.2.4 Business releases.	92
7.2.5 Change activities.	93
7.2.6 The IT release.	93
7.3 Loan Validation Service illustration	95
7.3.1 Creating the initial release	96
7.3.2 Creating the first IT release.	98
7.3.3 Validating IT Release 1	100
7.3.4 Completing IT Release 1	105
7.3.5 Creating Business Release 1	110
7.3.6 Changing Business release 1	112
7.3.7 Business Release 1 completion	115
7.4 Conclusion	117
Chapter 8. Deployment	119
8.1 Choosing a deployment strategy.	120
8.1.1 IT Centric deployments	120
8.1.2 Business Centric deployments	120
8.1.3 Two-phase deployment.	121
8.2 Difference between production and non-production deployment	122
8.3 Versioning policies	123
8.3.1 RuleApp and ruleset versioning policy	123
8.3.2 Running against multiple deployed versions	124
8.4 Managing resources	124
8.4.1 Managed Java XOM	125
8.4.2 Enabling XOM management in Decision Center.	125
8.5 Conclusion	126
Chapter 9. ODM DevOps	127
9.1 IT Centric DevOps.	128
9.1.1 Advantages	128
9.1.2 Disadvantages	128
9.1.3 IT Centric DevOps sequence	128
9.2 Business Centric DevOps	129
9.2.1 Advantages	130
9.2.2 Disadvantages	130
9.2.3 Regular cycle: Business change	130
9.2.4 Irregular cycle: Technical change	131
9.3 DevOps orchestration with UrbanCode Deploy	133
9.3.1 What is UrbanCode Deploy?.	133
9.3.2 The UrbanCode Deploy ODM plug-ins	133
9.4 Suggested APIs for DevOps	133
9.5 Conclusion	134
Chapter 10. ODM on Cloud	135
10.1 Considerations	136
10.2 IBM ODM components	136
10.3 User roles	137
10.4 Cloud environments	138
10.5 Workflow	140

10.5.1	Setting up the user roles	140
10.5.2	Collaborating to develop the decision service	140
10.5.3	Testing and promoting the decision service	141
10.6	IBM ODM on Cloud in a hybrid cloud environment	141
10.7	Conclusion	142
Chapter 11.	Branching and merging	143
11.1	Branching and merging to manage releases	144
11.1.1	Branching and merging	144
11.1.2	Snapshots	145
11.1.3	Naming conventions	145
11.2	Deployment events	145
11.2.1	Scenario A: Single executable service	146
11.2.2	Scenario B: Parallel executable services	146
11.3	Branching strategies	147
11.3.1	Main branch with snapshots only	148
11.3.2	Main branch with simple branches for changes	149
11.3.3	One branch per decision service version	151
11.3.4	Decision Governance Framework like branching	152
11.3.5	Advanced branching strategies	153
11.4	Example of branching and merging in IBM Operational Decision Manager	153
11.4.1	Leading practices	160
11.5	When to use branching and merging	161
11.6	Going from branching and merging to Decision Governance Framework	162
11.7	Conclusion	163
Chapter 12.	Conclusion	165
12.1	Where you go from here	166
Appendix A.	Acronyms	167
Related publications	169
IBM Redbooks	169
Other publications	169
Online resources	170
Help from IBM	171

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.


Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

Bluemix®
developerWorks®
Global Business Services®
IBM®

IBM UrbanCode™
ILOG®
PureApplication®
Redbooks®

Redbooks (logo) ®
UrbanCode®
WebSphere®

The following terms are trademarks of other companies:

Evolution, and Workforcebydesign are trademarks or registered trademarks of Kenexa, an IBM Company.

ITIL is a Registered Trade Mark of AXELOS Limited.

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

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

Preface

This IBM® Redbooks® publication introduces operational decision governance and describes in detail how to implement it using the IBM Operational Decision Manager (ODM) platform. ODM allows businesses to automate and manage day-to-day operational decisions. It provides an integrated repository and management components for line-of-business, subject-matter experts to directly participate in the definition and governance of rules-based decision logic, organized in decision services. Governance of changes to decision services is of particular importance and value.

This book describes how organizations can choose between the built-in ODM decision governance framework or a custom governance based on manually managed branches. Related topics, such as access control, permissions and user management, are covered and give a full view on decision service governance. You will find this book valuable if you are using or considering the usage of an operational decision management system in your organization, either with ODM on-premises or ODM on Cloud offerings.

This book was written to help assist the following target audience in applying Decision Management technology successfully:

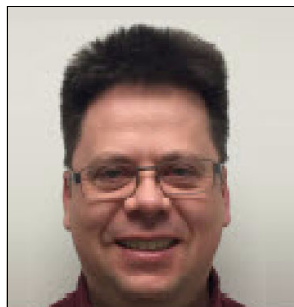
- ▶ IT Project Managers need to understand how decision governance differs from IT Governance, and how ODM straddles both worlds to facilitate agile change.
- ▶ IT Technical Architects need to understand how to architect ODM to sit inside both the IT and business worlds.
- ▶ Business Analysts need to understand the processes for changing business policies using ODM Decision Center.
- ▶ Business Rule Development Teams need to understand the best way to structure rule projects for scalability and maintainability.

Authors

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



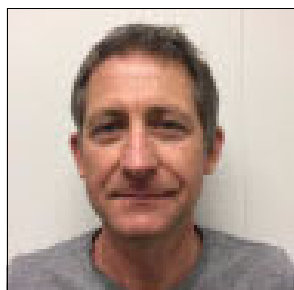
Jonathon Carr works in the IBM Cloud Client Technical Engagement specializing in Operational Decision Manager. He has over 15 years' experience in designing and developing Business Rules solutions and almost 20 years in IT. Jonathon has experience across all business sectors. Jonathon has helped build demos and present on ODM topics covering decision governance and big data. Jonathon has co-authored four IBM developerWorks® articles on Operational Decision Manager.



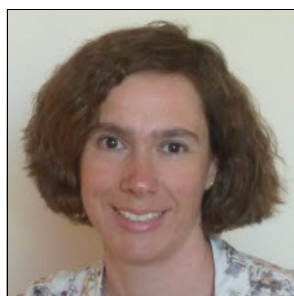
Eric Charpentier is an Operational Decision Manager (ODM) Level 2 Support Engineer with IBM Hybrid Cloud in Canada. Before joining the support team, Eric was an ODM Consultant and helped clients implement ODM in projects of all sizes and in many different industries. As part of the support team, he helps clients troubleshoot and resolve issues around ODM. Eric co-authored the first version of this IBM Redbooks publication, and has presented on multiple topics including decision governance. Eric has a Bachelors Engineering degree in Computer Engineering from École Polytechnique de Montréal and an MBA from HEC Montréal.



Nigel Crowther works in the IBM Cloud Client Technical Engagement specializing in Smarter Process (ODM, DSI). He has over 12 years' experience in designing and developing Smarter Process systems and 30 years in IT. Nigel has experience across all business sectors, with his greatest focus on finance. Nigel has written and presented on ODM topics covering rule governance, big data, and hot entities.



Guillaume Devillers is an Operational Decision Manager (ODM) Level 2 Support Engineer with IBM Hybrid Cloud in Germany. As part of the support team, he helps clients troubleshoot and resolve issues around ODM. Before joining the ODM support team, Guillaume was part of the BPM support team and an IBM WebSphere® consultant in France helping clients implementing BPM, WMB, MQ, and CrossWorlds in projects of all sizes and in many different industries. Guillaume has a doctorate in Plasma Physics from Université Joseph Fourier (Grenoble, France).



Sophie Fargier is a Level 2 Support Engineer for Operational Decision Manager (ODM), based in Paris. Over the past five years, she has been showing strong commitment in delivering excellent service to ODM customers. She helps them to configure, use, develop, and tune all components and versions of the ODM product family, keeping focus on customers' satisfaction. Prior to this position, Sophie worked as a Java Developer and as a Cross-brand IT Consultant in the IBM France Lab. Sophie has a Master Degree in Mathematics from University of Lyon I and an Engineering Degree from Télécom ParisTech in Paris, with master in computer science.



Julie Holm is a Business Policy Architect and Project Manager for Operational Decision Manager Services, based in Camp Hill Pennsylvania. She has been working in Expert Systems, Rule, Case and Frame based reasoning, and Decision Systems methodologies for twenty-five years, and has been with IBM ILOG® and IBM since 2006. She has helped a wide variety of customers implement Decision Management Systems including Federal Agencies, Health Care, Finance, Transportation, Insurance, and Banking customers, among others. She has been an analyst, designer, developer and rule author, has performed and managed testing at all levels, and has been a project manager and methodology author and has provided production support for a wide variety of Decision Management Systems. She also has a specialty in training and mentoring business users to realize the capabilities of decision systems and to maintain them with minimal involvement from IT. Julie has a Masters Degree in Information Systems from George Mason University with a concentration in Artificial Intelligence and Expert Systems.

This project was led by Margaret Ticknor, an IBM Technical Content Services Project Leader in the Raleigh Center. She primarily leads projects about WebSphere products, IBM PureApplication® System, and Cloud. Before joining the IBM Technical Content Services, Margaret worked as an IT specialist in Endicott, NY. Margaret attended the Computer Science program at State University of New York at Binghamton.

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

Antony Viaud

Offering Manager, IBM Operational Decision Manager, IBM Hybrid Cloud

Pierre Berlandier

STSM, IBM Operational Decision Management, Software Group Lab Services, IBM Cloud

Raffaele Cocucci

Business Analyst, IBM Global Business Services®

Callum Jackson

Integration Solution Architect, IBM Hybrid Cloud

Robert Prager

I/T Specialist, Global Business Services

Thanks to the following people for their support of this project:

- ▶ Ann Lund, IBM Redbooks Residency Administrator
- ▶ Russell Kliegel, IBM Redbooks Production Specialist

Thanks to the authors of the previous editions of this book.

- ▶ Authors of the first edition, *Governing Operational Decisions in an Enterprise Scalable Way*, published in 2013, were Pierre Berlandier, Eric Charpentier and Duncan Clark.

Now you can become a published author, too

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us.

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

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

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

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



Introduction

The purpose of this IBM Redbooks publication is to help architects, project managers, developers, and business analysts apply Decision Management successfully. *Decision Management* is the combination of tools, processes, and people that enables enterprises to respond quickly and cost effectively to business policy change.

Let us first examine the title of the book, *Governing Operational Decisions in an Enterprise Scalable Way*. What do the words *Governance*, *Operational Decisions*, and *Enterprise Scalable* mean?

► Governance

Figure 1-1 on page 2 shows the three levels of governance within the organization:

- The over arching definition is *Corporate Governance*, which defines the overall policies and ethics of the corporation.
- Underneath is *IT Governance* controlling application development using various methodologies, such as SCRUM, ITIL, or COBIT¹.
- The last is *decision governance*, which defines the methodology for changing business rules in the organization. It is decision governance that is the focus of this book.

¹ See COBIT official website: <http://www.isaca.org/Cobit/pages/default.aspx>

Figure 1-1 shows the governance hierarchy.

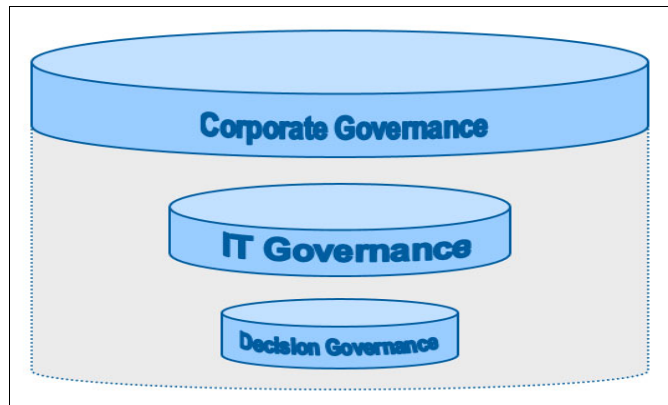


Figure 1-1 Levels of governance

- ▶ Operational Decisions

Operational Decisions are business policies controlling the enterprise. These policies are distilled into *Business Rules*, which are organized into *decision services*.

- ▶ Enterprise Scalable

Enterprise Scalable requires being able to govern changes in parallel over thousands of rules by hundreds of people. To do this, an enterprise-scale Decision Management System is required, such as IBM Operational Decision Manager (IBM ODM).

This chapter covers the following topics:

- ▶ Assumptions
- ▶ Decision governance
- ▶ The sample scenario
- ▶ Book layout

1.1 Assumptions

The focus of this book is on the latest features of IBM ODM V8.9.0.1 or later, decision services, Decision Engine, and the decision governance framework. The decision governance framework is an optional prescriptive workflow that enforces a process for changing rules within IBM Decision Center. This book assumes decision governance framework as the default approach to implementing decision governance. However, decision governance framework is optional, and it is up to readers to decide which approach works for them.

1.1.1 The ODM components

This book focuses on the key components of ODM Standard (Figure 1-2).

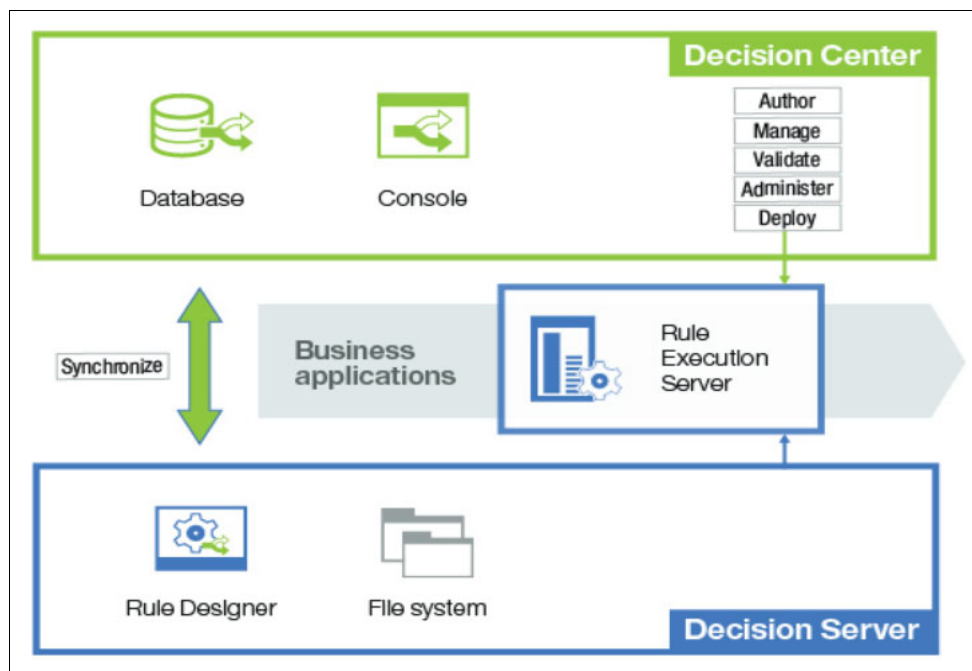


Figure 1-2 ODM components overview

The following components are shown in Figure 1-2:

- ▶ Decision Center, which has two consoles:
 - The Business Console enables business users to work inside a governance framework based on releases and change activities.
 - The Enterprise Console enables advanced features and is normally used by administrative users.
- ▶ The Decision Center database provides persistence for all decision artifacts.
- ▶ Decision Server includes Rule Designer and Rule Execution Server (RES).
- ▶ Rule Execution Server is the execution run time for business rules, and includes the RES console to enable administrators to deploy and manage decision services.
- ▶ Rule Designer provides developers a “one-stop” development environment to build all the artifacts and operations needed to create and maintain rule applications. Rule Designer is based on Eclipse. Artifacts are saved to the file system and committed to a source code control system.

You can find more detailed information about IBM Operational Decision Manager at the IBM Operational Decision Manager home page: [IBM Operational Decision Manager home page](#).

1.2 Decision governance

Decision governance defines the tools, processes, and people that facilitate business policy change.

1.2.1 The Benefits of decision governance

Figure 1-3 shows business policy changes being deployed more frequently than traditional application development releases. The result is agility. Agility enables the enterprise to respond quickly to change and gain a competitive advantage.

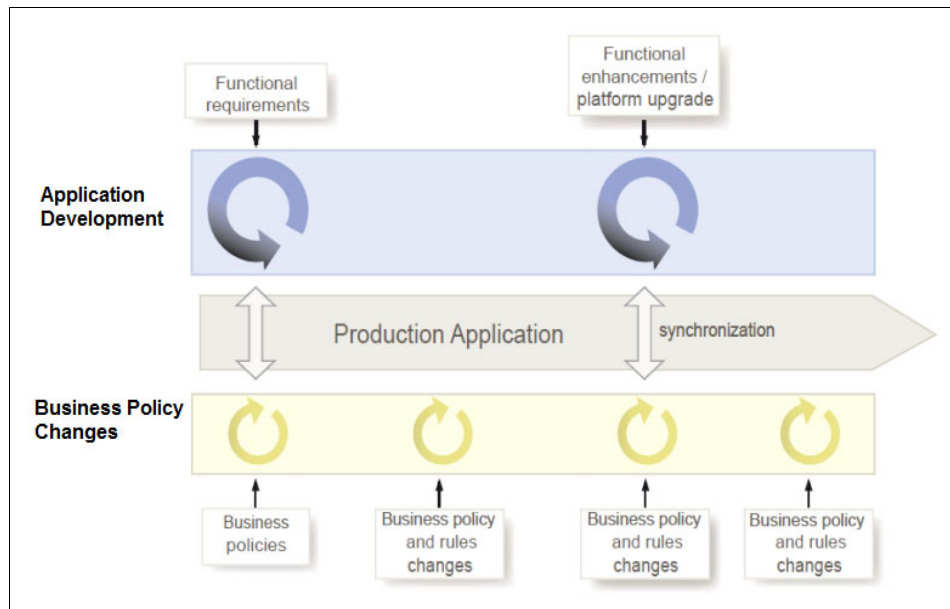


Figure 1-3 Business policy changes

1.2.2 The Evolution of decision governance

Figure 1-4 on page 5 shows the evolution of decision governance within the enterprise. Stage one shows traditional application development where business policies are scattered through the organization in code and spreadsheets. Changes to business policies are managed by the IT Governance process.

Stage two is to adopt ODM to encapsulate business policies into business rules. Business rules are developed, maintained, and tested using ODM, but changes are still managed within the traditional Application Development IT Governance process.

In stage three, business rules are maintained outside traditional Application Development. This is where the full power of ODM comes into play. The business takes control of the day-to-day agile changes to business policy. Changes to business policy are made safe and secure by the decision governance features of ODM.

The shift from phase one to phase three requires first the adoption of Decision Server to encapsulate the business policy, and then the adoption of Decision Center to enforce decision governance. The benefit of this effort is enabling agile change to business policy.

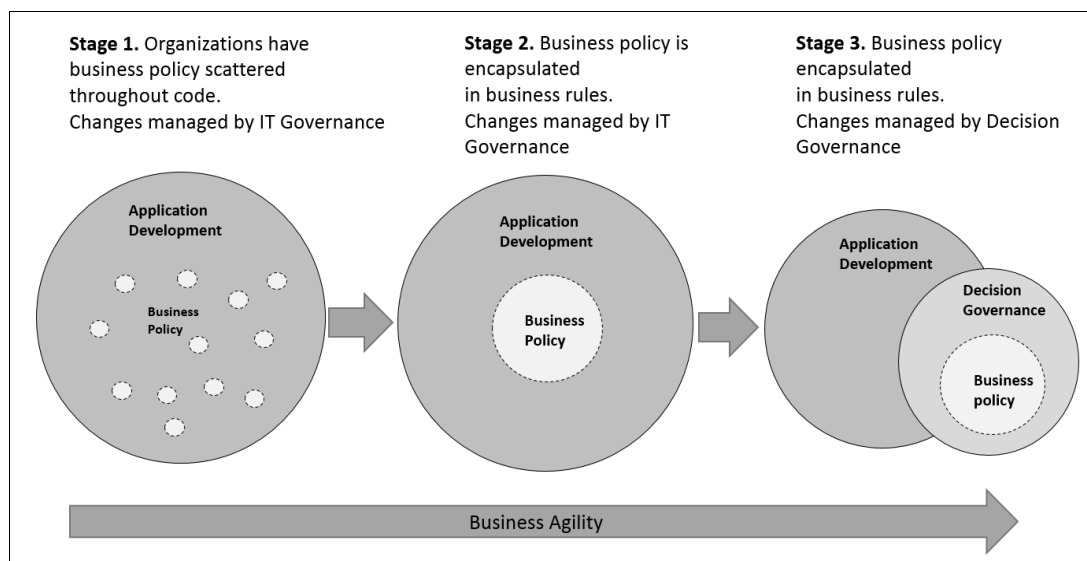


Figure 1-4 Evolution of decision governance

1.3 The sample scenario

This book uses the Loan Validation Service of a fictional banking organization to illustrate how IBM Operational Decision Manager can be used to govern decision making in an enterprise scalable way. This decision service is available within the sample server, so you can follow along and see in your own sample installation what is going on.

The goal is to show how the validation of eligibility for a loan can be acted upon in the following ways:

- ▶ Represented as a Loan Validation decision service
- ▶ Broken down into rule projects to support appropriate access to business users and IT roles
- ▶ Changed in an agile way by the appropriate roles with appropriate skill levels, ensuring traceability of changes to business policy
- ▶ Tested and validated to ensure that the changes meet the business requirements
- ▶ Deployed to the operational systems using the decision governance framework

1.3.1 Loan Validation background

The Loan Validation service provides a simple service to handle requests for large loans, like home mortgages. A borrower goes to the bank and applies for a loan for a specific amount. For the loan application, the borrower needs to provide information such as income, age, previous bankruptcy information, and so forth. The information from the application is used as input to the business rules to establish if the loan can be granted and under what conditions.

The bank uses IBM Operational Decision Manager to automate the loan decisions and return a decision. It validates input data, computes an internal score, determines eligibility, and returns the decision with appropriate comments. This helps the bank make quick, effective, and repeatable decisions for borrowers through a variety of channels. The use of ODM helps ensure that their lending policies are enforced, and allows them to make quick changes to the policies when they need to change, whether in response to changes to their own criteria or in response to changes in the legal requirements.

1.3.2 The Loan Validation solution

The bank's solution (Figure 1-5) consists of two components:

- ▶ Decision Management via Decision Center
- ▶ Decision Execution via Rule Execution Server

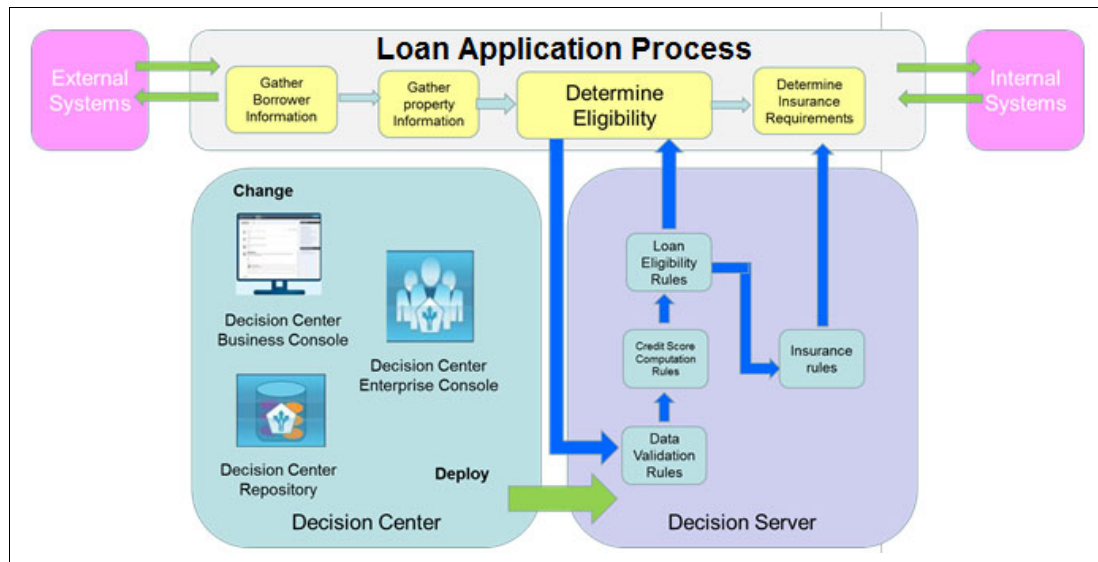


Figure 1-5 Loan Validation Solution

1.4 Book layout

This chapter outlines the main concepts and structure that this book will cover.

Chapter 2, “Decision governance for project managers” on page 9 plans for Decision Governance and provides an example to help plan for your projects. It also describes how to create a Center of Excellence.

Chapter 3, “Roles and responsibilities in governing decisions” on page 21 explains roles and responsibilities for governing decisions.

Chapter 4, “Securing the Decision Center” on page 31 explains how to create roles and apply users and security to Decision Center.

Chapter 5, “Designing decision services” on page 61 explains how to organize your rule projects for scalability and maintainability.

Chapter 6, “Processes” on page 73 explains the processes involved in delivering decision services, both from an IT and Business perspective.

Chapter 7, “Decision governance framework” on page 85 takes a deep dive into the Decision Governance Framework and shows how to use it for both simple and complex projects.

Chapter 8, “Deployment” on page 119 covers deploying decision services to the execution environment.

Chapter 9, “ODM DevOps” on page 127 provides DevOps guidance on automating the build and deployment of decision services, both from IT and business centric environments.

Chapter 10, “ODM on Cloud” on page 135 provides details on how to govern decision services on the cloud.

Chapter 11, “Branching and merging” on page 143 provides details on how to use the advanced repository management features of Decision Center when the decision governance framework is not used.

Chapter 12, “Conclusion” on page 165 concludes and summarizes the book.



Decision governance for project managers

Project managers need to ensure that decision governance processes and controls are in place as part of the project planning. They need to perform the following roles:

- ▶ Ensure that the project is staffed in a way that supports effective decision governance
- ▶ Plan for decision governance
- ▶ Manage risks and problems that occur within the process of decision governance

In addition to the project management for governance, this chapter deals with organizational change within the context of Decision Management Systems, and the role of a Center of Excellence (CoE) in managing that change.

The following topics are covered:

- ▶ Staffing the project team
- ▶ Planning for decision governance
- ▶ Project risks and problems associated with decision governance
- ▶ Changing your organization to realize your decision governance vision
- ▶ Conclusion

2.1 Staffing the project team

One of the most important parts of managing a decision-centric project is ensuring that the environment starts off right to develop a decision governance process. Key to this is paying attention from the beginning to the roles needed, and starting with some idea as to who will fill the roles during the staffing of the project.

On a project involving operational business decisions, the most critical roles to fill from the start are the IBM Operational Decision Manager (ODM) Architect and the Business Users. Ensure that you have people who know the business to fill the Business Users roles, especially SME and Business Policy Analyst (BPA). The ODM Architect is key to the Center of Excellence for ODM, and the Business roles are key to defining the requirements and the rules.

Not as critical, but still important is the role of Business Analyst. This is particularly true if the Business Analyst will do the authoring, which is a preferred practice for business rules. They will have a critical role in governance of the decisions.

Chapter 3, “Roles and responsibilities in governing decisions” on page 21 describes the roles and responsibilities associated with governance. Project Managers should be sure to read this chapter.

2.2 Planning for decision governance

There are two ways that organizations frequently develop decision governance plans and processes.

Some organizations, particularly ones that understand the decision paradigm and have strong business leadership, can develop their governance plan during the development of the first iteration of the decision services.

Many clients find themselves approaching governance separately after the first iteration of the rules. This can occur for the following reasons:

- ▶ This is the first time that the project team has worked with ODM, and they want to “walk before they can run”.
- ▶ The business is not ready to engage with decision governance.
- ▶ Governance for a new version of ODM is being put in place after a migration.

This chapter will present two options for decision governance planning:

- ▶ The implementation of governance within the implementation of the first iteration of the decision service.
- ▶ The implementation of governance as a separate iteration, preferably immediately after the first iteration is complete, but as a governance-only implementation.

Because the tasks are largely the same, they will be discussed only once.

2.2.1 Governance within the first iteration

The following section describes a high-level plan for a successful decision governance implementation, as implemented within the first iteration of the decision service. Some of the tasks run in parallel.

Table 2-1 presents both the first-iteration tasks and the governance tasks. Governance tasks are detailed in the dark blue boxes:

Tip: It is important to make decision governance one of the primary deliverables of the project. This is true whether the decision governance is done during the initial release, or as a separate release.

- ▶ Hold a decision workshop to bring IT and business together:
 - First the decision service itself is discussed, analyzed, and an initial design is done.
 - Second, governance for the decision service is discussed, analyzed, and designed. Discuss how decision services will be implemented and maintained by different teams.
- ▶ Define, build, and test the decision service and its surrounding application.
- ▶ In parallel, define, build, and test the decision governance process.
- ▶ Design, build, and test the infrastructure and deployment architecture.
- ▶ Train the business how to use Decision Center and how to create business releases.
- ▶ Go live with your first successful decision governance project.
- ▶ Create the Center of Excellence to promote decision governance to new areas of the business.

Table 2-1 Decision governance planning within the initial release plan

Task	Week																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Application Design Workshops	All Stake-holders																
Decision Governance Workshops			All Stake-holders														
Define Decision Service					BAs, Architects, and Developers												
Architect Solution						Architect Solution											
Build Object Model (XSD, Java)							Java Developers										
Test and fix defects in technical artifacts											Java Developers						
Define Business Rules							BAs, Rule Authors, ODM Architects, and Developers										
Define Decision Governance Process							BAs, Rule Authors, ODM Architects, and Developers										

Build Decision Service								BAs, Rule Authors, ODM Architects, and Developers							
Build tests and scenarios									BAs, Rule Authors, ODM Architects, and Developers						
Script Build and Deploy in All Environments					Script Build and Deploy in All Environments										
Provision Hardware (Cloud or On Premise). Install Product, enable firewalls, create users, roles and security.					Infrastructure Team										
Create test scripts for Decision Services					Test/QA Team										
Create Test scripts for Decision Governance process					Test/QA Team										
Test Decision Service											Test/QA Team				
Test Decision Governance Process											Test/QA Team				
Refine Decision Governance Process											BAs, Authors, ODM Architects, and Developers				

2.2.2 Governance tasks

This section describes the governance tasks required for the project.

Decision governance workshop

All of the business and IT stakeholders should be present for a workshop to learn about decision governance. They should discuss the organizational requirements for decision governance, determine requirements, and plan the full decision governance process. This is typically a series of discussions over a full week. The workshop is managed by the project manager but driven by the ODM and IT architects.

One concern is the time required for this process, particularly because if you are doing this within the first iteration, it will immediately follow the workshop defining the contents and scope of that iteration. However, leaving stakeholders out presents a significant risk to the development plan.

The decision governance workshop includes the following activities:

- Present the purpose and goals of decision governance. An individual with significant understanding of decision governance, either from within your organization or a consultant, presents to all stakeholders the unique needs, requirements, and goals of decision governance.

- ▶ Develop an organizational map. Define the current organization of the people who will be involved with the development and maintenance of the decision service. This group includes those who are involved with systems that interface with the decisions, people involved in quality assurance, end users, deployers, business users, and IT developers.
- ▶ Review the roles and responsibilities as described in Chapter 3, “Roles and responsibilities in governing decisions” on page 21. Overlay the decision governance roles and responsibilities on the organizational map.
- ▶ Describe the existing IT governance processes. Identify where the decision management needs can be integrated into the existing process. For example, many organizations have a requirement for regression testing. Regression testing for decisions may be understood as the same process as IT regression testing in your organization, which would mean that the only need in this is to provide new tools for decision testing.
- ▶ Identify and document new decision governance processes, including authoring, test, deployment, and execution processes.
- ▶ Identify supporting tools that will be needed for any of the governance processes. In addition to tools provided within ODM, you should identify any other tools that may be needed, or which are already used in your organization and would support your decision governance.
- ▶ Identify the expected organization of the decision logic, and define as project roles. For example, in our Loan Validation scenario, determining eligibility of the loan application may belong in one business organization, while a separate organization may maintain the logic for insurance requirements. These are *project roles*, as discussed in Chapter 3, “Roles and responsibilities in governing decisions” on page 21. For any projects that will be kept separate, identify who will be able to access that decision logic within ODM and who will not.
- ▶ Discuss versioning plans. Will there be interim releases between main releases? Are emergency releases possible? How often will the releases be made? Identify use cases for business policy changes, how business policy changes are initiated, and what the process for approving policy changes for implementation is.
- ▶ Discuss your process for deployment, including the details of approvals required for deployment, who will deploy (IT or the Business) in what circumstances, and how decision deployment will relate to deployment of applications interfacing with the decision services.
- ▶ Discuss whether your build and deployment will be automated (see Chapter 9, “ODM DevOps” on page 127) or whether it will be manual.
- ▶ Identify how ODM will enforce these roles, responsibilities, project roles, and versioning requirements. Discuss the decision governance framework if you are planning to use it.
- ▶ Document all of the decisions made, and ensure that all attendees agree.

Defining the decision governance process

As part of the workshop the decision governance process and how it integrates with existing IT governance processes is determined. The workshop includes assigning roles, permissions, and responsibilities. The team fully vets the process against existing IT change use cases for the policy to be realized.

Implementing and refining the decision governance process

The Business Users, ODM Architect, and ODM Rule Developers implement and test access and permissions within Decision Center, unit test the process for incorporating changes from Rule Designer (for technical changes) and a variety of releases and release combinations. These tests are generally done in a test environment or sandbox version of Decision Center and not in the production Decision Center environment.

Script build and deploy in all environments

The DevOps team (or a combination of Infrastructure and Development teams in organizations which do not have a dedicated DevOps team) prepares scripts for automatically building and deploying RuleApps. Build and deployment scripts can be developed in parallel to the decision services development.

Creating test scripts for the decision governance process

The primary purpose of the decision governance test scripts is to ensure that the decision governance process works correctly. Test scripts are created by an IT QA team, and their purpose is to verify the decision governance use cases identified during the workshop. They include the following test cases:

- ▶ Test users are assigned the correct roles.
- ▶ Are the users able to create releases, approve change activities, make changes to rule artifacts, test and deploy. Are they denied access when they should not be able to perform these actions?
- ▶ If using DevOps, test the build and deploy script.
- ▶ If manually deploying, ensure that only the appropriate users can deploy.

Using a second Decision Center in the test environment

Decision Center is usually installed in the production environment. However, for testing and refining the governance process, an additional installation of Decision Center in the test environment is a good practice. This allows the decision governance process to be tested before it goes into production.

However, do not make the mistake of continuing to maintain rules in the test Decision Center after you go “live”. It is there only for testing and once fully tested, users should move to the production Decision Center. The production environment is where all real decision authoring should be performed. Occasionally, the decision services from the production environment can be brought back to the test environment if the decision governance process needs refining and retesting.

Tip: Plan for two installations of Decision Center, one in a test environment and one in a production environment. This will allow you to accomplish the following tasks:

- ▶ Verify governance processes and permission management
- ▶ Perform and verify upgrades of ODM outside of production

2.2.3 Governance only process

Table 2-2 on page 15 presents the timeline, in a similar format as Table 2-1 on page 11, for a situation where the governance work is done outside of a project iteration. This frequently happens when an organization is new to decisions, and needs to go through the process of a first iteration. It can also happen when an organization is migrating to a new version of ODM with a significantly different governance framework.

In this case the tasks are basically the same as described previously. However, there will not be decision development tasks interspersed between the governance tasks.

Table 2-2 Decision governance planning for governance only iteration

Task	Week									
	1	2	3	4	5	6	7	8	9	10
Decision Governance Workshops	All Stake holders									
Define Decision Governance Process			BAs, Authors, ODM Architects & Developers							
Script and Build Out Decision Governance					DevOps team - OR ODM Developers					
Create users, roles and security for Decision Governance					DevOps team - OR ODM Developers					
Create Test Scripts for Decision Governance Process			Test/QA Team							
Test Decision Governance Process								Test/QA Team		
Refine Decision Governance Process									BAs, Authors, ODM Architects & Developers	

Note: The Governance only process is shorter in time, because it is assumed that resources in this scenario are fully invested in governance, rather than being shared with the development team. The person-hour effort is similar, and depends on the complexity of the task. Complexity cannot be estimated until after the workshops are complete.

2.3 Project risks and problems associated with decision governance

An important part of managing a decision management system (DMS) project is associated with managing the risks and issues. The following risk areas impact the project from a decision governance point of view. If they are present, they should be identified, evaluated, and tracked within the governance portion of the project:

- ▶ Lack of communication between business and technical teams, or lack of access to knowledgeable business people.
- ▶ Lack of ODM Skills.
- ▶ Lack of Business Analysis, SME and Business Policy Analyst Skills.
- ▶ Weak project organization.
- ▶ Inadequate training or preparation in Business Rule paradigm or approach (resulting in a lack of understanding of the separate roles of business and technical teams).
- ▶ Ineffective or incomplete rule testing strategy.

- ▶ Overly complex domain model, particularly if it includes a significant portion of information the rules do not use (often occurs when an organization implements an enterprise model as a domain model).
- ▶ Implementing decision governance in an organization with weak IT governance procedures requires additional training and buy-in, especially from technical teams. There is a significant risk that the IT organization will not see the advantage or need for decision governance in this situation.

2.4 Changing your organization to realize your decision governance vision

It is frequently necessary to make changes within an organization to realize your decision governance vision. Sometimes this is only to accommodate new roles, but at times it is necessary to more intentionally organize around the decision governance process. This section describes how changes within an organization impact governance and how they can be handled.

Here are some steps to help you get started:

- ▶ Start by identifying the specific governance roles that your organization needs, including the responsibilities that these roles have. This is your governance vision.
- ▶ Map the governance roles that you identified to functional roles within your organization identified in your organization map. This action might possibly involve identifying specific people to fill a governance role. This is your governance reality.
- ▶ Identify the gaps in people or skills as well as the changes that are required to move towards your governance vision. Here are some possible changes:
 - Temporarily use external consultants to disseminate the required skills through your organization.
 - Training people (formal or on the job) to give them the skills they need. Use IBM approved on line or in class courses for this. (See reference section “Online resources” on page 170).
 - Hiring new people to complement your existing team.

Tip: Identify gaps in people or skills as well as the changes required to move towards your governance vision.

Before you attempt any organizational changes, it is important to consider the emotional response of the people who might be affected by such a change. The organizational change that is required for supporting operational decision governance and management may not be extensive, especially compared to other types of organizational changes, but it does have an impact. With this situation in mind, you should focus on two fundamental points when you consider this specific type of organizational change:

- ▶ Develop your workforce
- ▶ Communication is key

Workforce

As the organization works towards supporting operational decision management, the future needs of the organization are more clearly defined and the following questions should be answered:

- ▶ What are the new roles and their responsibilities?
- ▶ How many people are required to fill each role?
- ▶ What knowledge of your existing business logic is required for each role?
- ▶ What are the skills that are required for each role?

To answer these questions, start with a small subset of key stakeholders that can provide their input and participate in constructive discussions around the organizational changes, what the changes mean to the other stakeholders, and to make sure that most potential roadblocks are identified and mitigated early on.

The answers to the previous questions then let you identify in more detail how the gaps to achieve the goals of the organizational change can be filled. To summarize, it is important to perform the following tasks:

- ▶ Identify specific employees who play the roles that are identified.
- ▶ Identify skills or knowledge gaps and create development plans for each employee.
- ▶ Identify training and education that might be required to help employees achieve the goals.

The key stakeholders can help provide details about all this information and prepare the material that is required to present to all the stakeholders in follow-up meetings. These meetings are important in supporting your communication goals for the organizational change.

It is important to privilege business knowledge over existing technical skills, particularly in identifying Decision Authors, Decision Validators, and Decision Testers. Operational decisions center on your existing and future Business logic, and require a deep operational understanding of that logic. It is usually easier, and often better, to train someone who knows the business deeply to write business rules than it is to bring someone with a technical background to understand the business.

Also, someone with roots in the business will develop better vocabulary, better presentations, and more maintainable, simpler rules. Technical resources cross trained to understand the business inevitably bring their technical jargon and understanding to the task, which is not good for business rules. Organizations that feel that their business is incapable of taking over the rules, inevitably have rules written by programmers rather than business experts. Business people also have a natural level of ownership of the business logic, which helps provide buy-in to the changes.

Communication

Because you might encounter some resistance from some employees, it is important for you to address this resistance quickly and help them align with the vision and direction the organization is taking. To achieve success, it is critical that communication with stakeholders take place early so that they are made aware of the change and provided with information about the change. With time, they start understanding and accept the change before they take ownership of the change and commit to it.

Here are some key points to remember:

- ▶ Communicate the vision of the change early by providing details about the benefits and impact:
 - Reasons for the change
 - Benefits for them and the organization
 - Details of the change (when, where, and who is involved)
- ▶ Communicate often and repeat the message through multiple channels
- ▶ Involve the stakeholders by having an open dialog with them

The changes might be as simple as identifying the roles and responsibilities and assigning them to specific people in your organization. In that case, the communication may be achieved by calling a meeting with all the stakeholders that are involved and discussing with all of them the roles and responsibilities that are required, reviewing them, and assigning them to the people in the room. Then, the details can be discussed and clarified. Should you take this approach, be ready to discuss the topics about the workforce development as detailed previously.

2.4.1 The role of the Center of Excellence

It is important to highlight the potential role a Center of Excellence (CoE) has with helping an organization through a change, and with helping the people build the skills and knowledge they need to fill their new roles. While the work of a Center of Excellence is not limited to Governance, it can support the decision governance process as well as other best practice processes in relations to your decision services.

Tip: The CoE does not have to be an official office, at least not at the beginning. It can simply be a group of practitioners sharing their experiences with others.

The CoE can take a simplified form at the beginning where it simply consists of a community of practitioners within the organizations where the participants share their experiences, practices, and guidance with others.

This community can play an important role in the success of decision management and governance in your organization. At first, the participants are the people that run the first project in your organization, but as the usage of decision management continues to expand, these people possibly become the experts in the community.

The CoE community should help with the following tasks:

- ▶ Acquiring and keeping required skills so that the skills are available for other projects internally. This task allows for sharing of resources among multiple projects.
- ▶ Developing a broader set of people with the required skill base by supporting new people in developing their skills and knowledge.
- ▶ Providing coaching and mentoring for new people by giving them with a way to communicate with more experienced people.
- ▶ Documenting good practices and methodologies in your specific organization.
- ▶ Providing a first point of contact for people to discuss issues or challenges they are facing.

It is important to make sure that there is some enticement tied to the participation in the CoE and that the time spent by participants contributing to the community might have to be absorbed by each project. But this action might be a critical success factor for using decision technology in your organization.

The building of a CoE might require external expertise to kick-start the process and to get the CoE established. As it is built, the internal CoE staff should acquire the knowledge and skills required to keep the CoE running after the external resources leave. The CoE should also capitalize on the experience and practices of the first few projects as a starting point for future good practices, and practices to avoid!

Tip: If ODM skills are lacking it is a good idea to bring in temporary experts to kick-start your organization's CoE.

The evolution of the Center of Excellence

With time, the initial CoE can evolve into more:

- ▶ Project people become CoE champions to support the next project.
- ▶ The participation in the CoE community is essential for gathering common knowledge.
- ▶ Eventually, there might be a need to fund a specific role to lead the CoE.
- ▶ Champions can eventually evolve into the next level as *internal consultants* as part of the CoE.
- ▶ The CoE must be adapted so that it fits with your processes and evolves over time.

Much more could be said about the topic of the Center of Excellence, but it falls out of the scope of this IBM Redbooks publication.

2.5 Conclusion

In this chapter, we provided a high-level view of the implementation of governance from the project manager's view, with some details as to the tasks that would be typically encountered in the implementation project.

Specifically, we noted the following ideas:

- ▶ Governance Development Activities can be planned into an initial release, or into a separate technical release.
- ▶ The identification of appropriate roles and responsibilities, especially on the business side is a crucial component of the management of a decision services project.
- ▶ Managing change is as important as managing project tasks, and a Center of Excellence can be a valuable tool for doing that.



Roles and responsibilities in governing decisions

Decision governance involves many people at many different stages, usually at least some from business and technical areas. Stakeholders can involve people from all over the organization to identify business rules; write, review, and test business rules; deploy business rules; and maintain business rules. This chapter provides information about the following functions:

- ▶ Identifies the typical roles within an organization
- ▶ Describes how to identify these roles in your organization
- ▶ Addresses the role of other stakeholders
- ▶ Explains each role's involvement in effective and efficient decision governance

One of the goals of decision governance is to support the organizational changes that are required to run the governance processes. One of the first steps towards this goal is to identify the roles and responsibilities that are required to support and run the processes.

This chapter defines governance roles and responsibilities and provides guidance for organizational changes.

This chapter covers the following topics:

- ▶ Decision governance roles
- ▶ Functional roles and responsibilities
- ▶ Secondary roles and responsibilities
- ▶ Clarifications about roles
- ▶ Conclusion

3.1 Decision governance roles

Decision governance was introduced in Chapter 1, “Introduction” on page 1. This chapter highlights *governance roles* involved in decision management.

A role is not equal to a position or a job title. A specific person, who holds a specific position in an organization with a job title, might play multiple roles. Similarly, a single role might end up being played by multiple people or a team. To add an additional layer of complexity, the same person might perform different functions within various contexts.

A governance role has the following characteristics:

- ▶ It is a set of defined and related responsibilities.
- ▶ It is typically performed by people, although some portions can be automated.
- ▶ It is required for operational decision governance and management to be successful in an organization.
- ▶ It should *not* be confused with an organizational position that is the job title and function of a specific individual in the organization.

Experience scale: Some of the roles that are described in the following tables have different levels that are based on the maturity and experience of the person that plays that role. It is easy to make the parallel to functional positions that are advertised as junior, intermediate, or senior positions. Similarly, you can fully expect this type of experience scale to be applied to some of the governance roles that are listed.

The following tables describe the roles and the responsibilities associated with them. The *Also known as* column provides some additional names that are sometimes used. Primary roles are held by people who are using IBM Operational Decision Manager (ODM) directly, and are described in section 3.2, “Functional roles and responsibilities” on page 23. Secondary roles are held by people who are not using ODM, but who impacts the governance process for the rules and are described in section 3.3, “Secondary roles and responsibilities” on page 26.

Primary roles generally will be codified within the permissions and access controls within ODM, while secondary roles will either have no permission and access, or will only have read access within ODM. This does not mean that secondary roles are not important, only that people in secondary roles will not be directly developing, maintaining, or deploying rules using ODM.

These roles are examples only and are not meant to be an exhaustive list that might be required for your organization. This list is a starting point and organizations should adjust and augment this list to fit their specific situation. The list of applicable roles evolves with time as the processes themselves evolve.

Not all the roles are needed in every installation. In the lists below, *core roles* are identified. Every organization should have every core role covered (even if multiple roles are covered by a single person.) Most organizations should consider all of the roles.

3.2 Functional roles and responsibilities

This section covers the functional roles and responsibilities.

3.2.1 Business roles

Business roles, detailed in Table 3-1 and Table 3-2 on page 24 are roles covered by the business people who interact directly with ODM. One of the great values of ODM is that it allows business people to develop, validate, test, maintain, and in some cases deploy business rules directly without being dependent on an IT organization. In the best case, these roles are assigned to the business people who know the business best, often the people doing the actual work of the business.

Table 3-1 Business roles in IBM ODM governance

Governance role	Responsibilities
Release Manager *Core role	Works primarily in the Business Console component: <ul style="list-style-type: none">► Orchestrates the lifecycle of a decision service, and is ultimately responsible for the deployment of a decision service release to production.► Follows a staged progression from development to test and production.► Creates development branches or releases:<ul style="list-style-type: none">– Defines change and validation activities for rule developers, business users, and integrators.– Assigns ownership of work, reviews, and approvals.► With author rights, the person in this role can create deployment configurations in the Business console for the development, test, and production environments.
Business User	Works primarily in the Business Console component: <ul style="list-style-type: none">► Implements and maintains some or all of the business rule artifacts that are in a decision service.► Runs functional tests and simulations in the development environment to validate the changes that are made for a release.► Can deploy a decision service to the test environment to validate changes in a test application.► Can also deploy to the development environment.► Can participate in the review or approval process for other business-user or integrator activities.

3.2.2 Business users

The role of business users can be implemented as a very simple, single role, and usually is when organizations begin using ODM. Also, in an IT-centric organization, simple business roles are often sufficient, and are often sufficient when working on the Cloud. More complex and more mature organizations often want to further decompose “Business Users.” Table 3-2 describes a number of core functional roles that provide a more complex view of Business Users.

Table 3-2 Business user role details

Governance role	Responsibilities	Also known as
Business Policy Analyst *Core role	<p>Manages the implementation of a business policy:</p> <ul style="list-style-type: none"> Assists the business in identifying existing decisions and gathering the required supporting documentation around them. Formalizes the business terminology (vocabulary) used in Operational Decision Manager. Creates and manages the domain model that represents the business entities and attributes and their relationships. Designs the structure of the implementation to support the design requirements and evolution of the decisions (shared with Rule Developer). Acts as a liaison between the business and IT at the time of the design and implementation of the decisions. Where Decision Modeling Notation is used for rule discovery, the Business Policy Analyst creates the DMN. 	<ul style="list-style-type: none"> Business Policy Architect Rule Analyst Rule Architect
Rule Author *Core role	<p>Authors rules to support decisions:</p> <ul style="list-style-type: none"> Manages and runs test cases to unit test the decisions they have authored. 	<ul style="list-style-type: none"> Author Rule Writer Business Rule Author
Rule Tester *Core role	<p>Manages and runs test cases to test the decisions:</p> <ul style="list-style-type: none"> Might review and approve rules to assure that they meet specific business guidelines. Runs tests and simulations. 	
Rule Validator	<p>Reviews and approves rules to assure that they meet specific business guidelines, often legal or high-level policy guidelines:</p> <ul style="list-style-type: none"> Might or might not manage and run test cases to support validation only. 	Approver
Business Analyst	<p>Gathers requirements from the business for technical artifacts to support Rule authoring:</p> <ul style="list-style-type: none"> Assists the Business Policy Analyst in creating and managing the domain model and mapping it to organizational data. Assists the Business Policy Analyst in designing the structure of the implementation, in particular identifying requirements for integration and transformation of external systems to support the rule implementation. 	

3.2.3 Technical roles

Technical roles, detailed in Table 3-3, are roles covered by the technical people who interact directly with ODM.

Table 3-3 Technical roles in ODM governance

Governance role	Responsibilities	Also known as
Rule Developer *Core role	Designs the structure of the implementation to support the design requirements and evolution of the decision services (shared with the Business Policy Analyst): <ul style="list-style-type: none">► Develops technical functionality to support authoring (XOM and utility functions).► Creates the initial version of the business rule artifacts, including technical rules, decision tables, and ruleflows.► Runs the decision service, locally or in the cloud development environment, until adequate results are achieved.► Publishes the decision service from Rule Designer to Decision Center.► Collaborates with the release manager and business users in authoring and governance activities.► Collaborates with the Integrator to integrate the decision service into an application.	
ODM Decision Center Administrator *Core role	Initially configure and administer Decision Center: <ul style="list-style-type: none">► Creates and configures projects.► Can also play the role of Permission Manager.	rtsAdministrator
Permission Manager	Works primarily in Business Console: <ul style="list-style-type: none">► Implements the security policy on decision services.► Creates groups, sets the permissions, adds users to the groups, and sets the groups on decision services.	
ODM Decision Server Administrator	Administers the Decision Server after the release has made it to production.	resAdministrator
Deployment Manager	Configures the deployment configuration within Decision: <ul style="list-style-type: none">► Center or Rule Designer (for example the decision service) packaging to deploy to Decision Server.	rtsConfigManager
Deployer	Can deploy decision services in all environments. Generates API descriptions from the Rule Execution Server (RES) Console, for easy publication and management of runtime decision services with API management solutions.	resDeployer

Governance role	Responsibilities	Also known as
ODM Architect *Core role	Designs the solution to ensure the performance of the rule execution and usability of the Decision Management platform: <ul style="list-style-type: none"> Plans and designs the infrastructure and deployment patterns of Operational Decision Manager in the enterprise environment to support the requirements and evolution of the decisions. Supervises the development efforts. Requests technical changes to support business decision changes (new data sources and XOM changes). Provides technical approval for a release. 	
IT Integration Developer	Uses the Business Console component in addition to other development, integration, and test tools: <ul style="list-style-type: none"> Builds the applications that call a decision service in the development, test, and production environments. Can be involved in the validation activities that are defined by the release manager. Can deploy decision services to the development and test environments. Can view and use the decision services that are deployed in the production environment. Develops components around integration to other systems, web services, and data services. 	Integrator
IT Quality Assurance (QA)	Performs end-to-end integration testing.	Tester

3.3 Secondary roles and responsibilities

This section covers the secondary roles and responsibilities.

3.3.1 Secondary business roles

Secondary business roles are composed of people whose interest is in the business logic of the organization, but who are not direct users of ODM, as shown in Table 3-4. These are often business stakeholders in the decisions.

Table 3-4 Secondary business roles for ODM governance

Governance role	Responsibilities	Also known as
Business Owner *Core role	Requests business decision changes: <ul style="list-style-type: none"> Prioritizes decision change requests in an agile environment. Provides business approval for a release. 	Policy Owner
Change Control Board (CCB)	Triages and selects change requests: <ul style="list-style-type: none"> Prioritizes change requests, or selects change requests to include in a specific release. Delegates the delivery of a release to a release manager. 	Steering Committee

Governance role	Responsibilities	Also known as
Subject Matter Experts (SMEs) *Core role	Provides Business knowledge about the subject matter of the decisions, if they are not Rule Authors.	

3.3.2 Secondary technical roles

Secondary technical roles (Table 3-5) are held by individuals who have a technical interest in the rules, but who are not working directly with ODM artifacts. These roles are required for the touch points between the decision governance and IT governance. These roles are sometimes collectively referred to as *IT Infrastructure Administration*, and are in the rest of this IBM Redbooks publication.

Table 3-5 Secondary technical roles for ODM governance

Governance role	Responsibilities	Also known as
LDAP Administrator	Creates and maintains users and groups in a user and group registry, such as an LDAP server to support the security requirements.	
Database Administrator	Participates in the installation of ODM: <ul style="list-style-type: none"> ► Supports updates to the ODM Databases. ► Creates and maintains databases as required by ODM. ► Works with the Business Policy Analyst to ensure that information needed by the rules is available and to define where to get needed information. 	
Application Server Administrator	Supports or performs initial installation of ODM: <ul style="list-style-type: none"> ► Creates and maintains users and groups to support the security requirements. ► Deploys the ODM EAR files. ► Maintains application infrastructure for applications integrated with ODM. 	
Networking Administrator	Supports or performs initial installation of ODM: <ul style="list-style-type: none"> ► Configures DNS. ► Configures firewall security. 	
IT Architect	Designs the solution to ensure the quality of the overall system: <ul style="list-style-type: none"> ► Defines the overall system. ► Architects and designs the infrastructure external to ODM, or into which ODM will be placed. ► Provides technical approval for a release from an overall technical perspective. 	<ul style="list-style-type: none"> ► Software Architect ► Application Architect

3.4 Clarifications about roles

Here are some specific comments about the roles:

► *Business Policy Analyst versus Business Analyst:*

Where both roles exist, their functional differences are connected to the difference between the decisions and the technology that supports the decisions:

- Business Policy Analysts concern themselves with the decisions, the information needed to enable those decisions, the structure of the decisions and the projects, and the best practices associated with those decisions. Business Policy Analysts might also perform the business analysis functions.
- Business Analysts are a more conventional role, which identifies requirements for technical teams. Business Analysts are involved when there are specific requirements that must be addressed outside of the rules, by technical members of the team. As they become more familiar with decisions, the decision process, and the requirements of decisions, Business Analysts can move into the role of the Business Policy Analyst.

► *Subject Matter Expert versus Business Policy Analyst:*

In the context of decision management, a Business Policy Analyst often discovers and provides the domain knowledge that is required for the project, and in some cases acts as a liaison to subject matter experts. This does not mean that on a project you might not have a subject matter expert that is available, but from experience, SME availability is often a challenge to projects.

► *Subject Matter Expert versus Rule Author and Rule Tester:*

Because SMEs are the people doing the work, and the people who understand the business rules well, they are an excellent choice for Rule Authors. As noted previously, SME availability can be a challenge, but in situations where they can be dedicated to a rule project, they are extremely valuable Rule Authors, particularly if they are well-mentored and supported.

Likewise, if they cannot be dedicated to a project, SMEs are excellent Rule Validators. They can provide high-quality test data that reflects their real experience, and can review rules and see business issues in them easily. If your organization permits the dedication of some of your SMEs to a rule project, that is a powerful positive for the quality of your decisions.

► *Rule Author versus Rule Developer:*

A Rule Author focuses on authoring rules through the Business Console of the Decisions Center. The Rule Developer is technical, works both with the Rule Designer and Business Console to develop every rule project artifact that is required to run the rules.

3.5 Conclusion

This chapter described the roles and responsibilities that are related to decision governance. The chapter provided a simple set of roles and responsibilities that can be used as a starting point by organizations, and some of the options for more complex business user roles. It provides a process of identifying and assigning roles:

- ▶ Map your existing organizational roles that will interact with the decisions and decision governance.
- ▶ Identify both functional and project roles for your decision services, using the lists of suggested primary and secondary roles.
- ▶ Identify the gaps in people or skills and appropriately map the decision governance roles to your organization.

After you identify the roles and responsibilities for your decision services in your organization, you have an initial basis to perform user management tasks.



Securing the Decision Center

The objective of this chapter is to describe the Decision Center security configuration. It provides information about the user management to access Decision Center (authentication) and artifact permissions (authorization) within Decision Center.

We first describe the role of the three components involved in Decision Center security:

- ▶ Predefined Decision Center UI roles
- ▶ The Decision Center groups
- ▶ The decision governance framework

We then suggest three sample permission patterns. For each pattern, we outline the security configuration and for the third pattern we include details about how to use a Lightweight Directory Access Protocol (LDAP) registry.

This chapter covers the following topics:

- ▶ Authentication
- ▶ Access control levels
- ▶ Permission patterns
- ▶ Security configuration process
- ▶ Sample team
- ▶ Simple pattern
- ▶ Fine-grained pattern
- ▶ Project pattern
- ▶ Conclusion

4.1 Authentication

Authentication is the primary level of security.

Authentication is the responsibility of the application server and is configured by the Application Server Administrator.

4.2 Access control levels

Authorization is the secondary security level establishing which features of Decision Center a user can access once logged in. This access is determined by the role of the user and what permissions the user is granted.

It is strongly advised to delegate authorization to the Decision Center application. This allows for management of users, groups, and roles within Decision Center itself, and gives the team managing the decision services more flexibility and control. For more information see IBM Knowledge Center about Operational Decision Manager (ODM) ([Step 3: Configuring access to the Decision Center](#)).

Alternatively, the application server may be used for both authentication and authorization and this feature remains available for upward compatibility when migrating from previous ODM versions. More details on this can be found IBM Knowledge Center about Operational Decision Manager (ODM) ([Access with authorization from the application server](#)).

The remainder of this chapter assumes that application server security is essentially used to authenticate users while, following our suggestion, Decision Center is delegated custom authorization configuration.

Decision Center users' effective privileges on each artifact can be represented as a Venn diagram, shown in Figure 4-1.

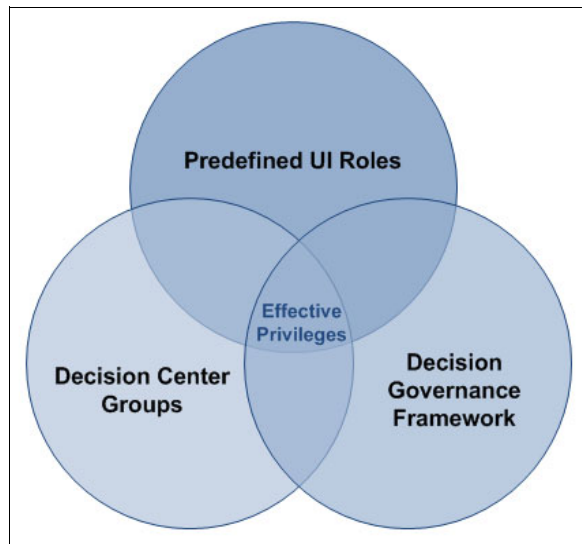


Figure 4-1 Decision Center effective privileges Venn diagram

The three levels of user access combine to give the user's effective privileges:

- **Predefined Decision Center UI roles**

Users must have at least one of the four predefined Decision Center UI roles to access Decision Center. A UI role determines what parts of the Decision Center UI are available to an authenticated user.

- **Decision Governance Framework State and Governance roles**

If you are using the decision governance framework (Chapter 7, "Decision governance framework" on page 85), access to the rule artifacts is also determined by the governance role that you have been assigned, and the state of the release and activity. For example, if a release is completed, the user cannot change any release artifact, even though the user might have full access from their roles and groups.

- **Decision Center groups**

User groups can be used to control access to decision service projects. If Decision Center project security is enabled on a decision service project, then Decision Center groups are granted view, create, update, and delete permissions on the different types of rule artifacts. Decision Center groups are defined in the ADMINISTRATION tab of the Business Console, and decision service artifact permissions are configured from the Enterprise Console.

Calculating the overall access permissions of an artifact is governed by the following three principles:

- **Decision Center Group Permissions Order**

Within a Decision Center group, the order in which the permissions are defined is important. The permissions are applied from top to bottom, so entries lower in the table override conflicting entries defined higher up in the table.

For more information, see IBM Knowledge Center, [Overview of permissions](#).

- **Combining Decision Center group permissions**

If a user is a member of more than one of the Decision Center groups that can access a given release, Decision Center merges the permissions. If there is a conflict Decision Center chooses the least restrictive group.

For more information, see IBM Knowledge Center, [Overview of permissions](#).

- **Aggregate the Access Control Level permissions**

If access to an artifact is denied at one level, it can't be overridden and granted by another level. Therefore, for an artifact to be accessible, access permission must be granted at all three levels (predefined Decision Center UI roles, decision governance framework roles and states, and Decision Center groups). Artifact permissions are formed by a logical AND.

4.3 Permission patterns

We describe three permission patterns applicable to the decision governance framework that can get you started quickly. We suggest that you choose the pattern that best fits your organization. You may wish to combine aspects of the different patterns to come up with your own custom security access model:

- Simple pattern (less than five team members)

Only coarse-grained roles for Administrators and Users are needed. Users have full power to view, create, update and delete all artifacts of all decision services. Administrators do the admin-related tasks. Works well for a small team that manages all rules.

- Fine-Grained pattern (less than 10 team members)

Fine-grained roles enable fine-grained access and artifact permissions. Works well for a small-medium sized team that manages all rules. Team members have different skill sets and responsibilities.

- Project pattern (greater than 10 team members)

Same as the fine-grained pattern, but different teams manage different decision services. Works well for an organization that has strict separation between departments requiring specific teams to manage their own set of decision services.

Each of these patterns is an extension of the previous one, which enables progressive implementation of the security policy as ODM coverage within the enterprise grows, as shown in Figure 4-2.

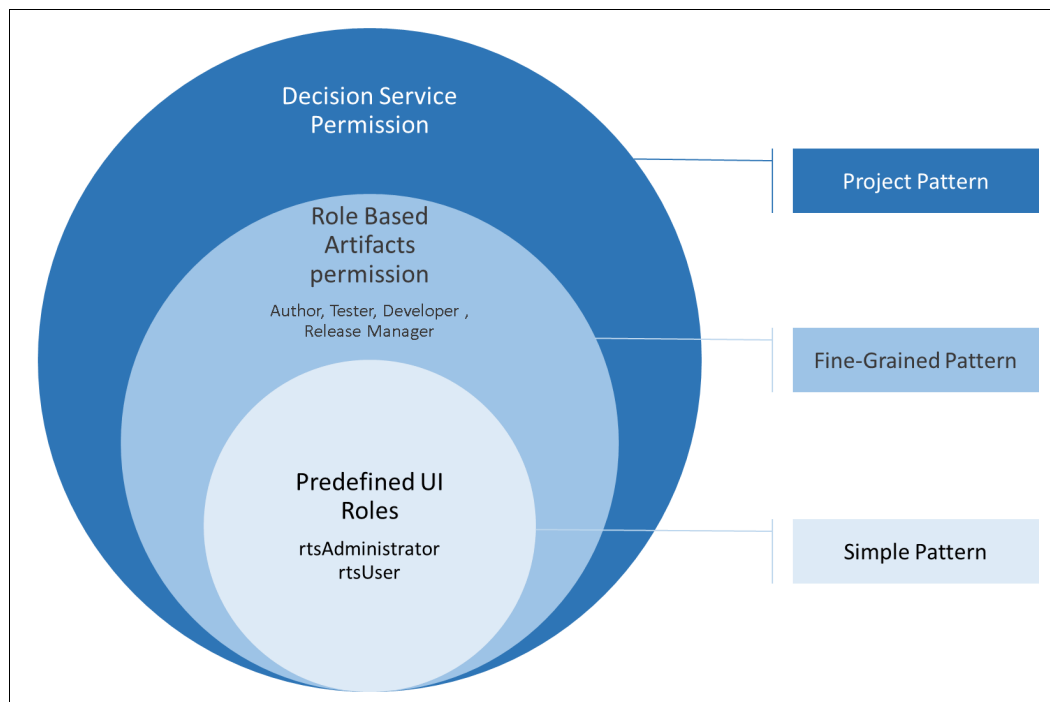


Figure 4-2 Decision Center permission patterns

4.4 Security configuration process

Depending on the User Management pattern, the overall user management configuration process is made up of the task list shown in Table 4-1.

Table 4-1 User Management configuration tasks

Number	Admin	Task	Simple	Fine Grained	Project
1	LDAP	Define user group	Y*	Y*	Y*
2	Application Server	Configure security to rely on LDAP	Y*	Y*	Y*
3		Define application server users and groups	Y**	Y**	Y**
4		Configure Decision Center application group to role mapping	Y	Y	Y
5	ODM	Define LDAP connections	N	Y*	Y*
6		Pull LDAP users and groups	N	Y*	Y*
7		Manually define Decision Center users and groups	N	Y**	Y**
8		Enforce decision services project security	N	Y	Y
9		Configure project permissions	N	Y	Y
(Y=Yes apply to the pattern, Y*=Yes if LDAP, Y**=Yes when no LDAP, N=No, does not apply)					

Some tasks are optional and depend on the usage of an LDAP registry (Y*) while others become necessary when LDAP is not involved (Y**).

When illustrating the security configuration process for each pattern, the tasks in bold will be described. An LDAP registry will be added to the landscape in the Project pattern.

We will be demonstrating the application server security of each pattern using IBM WebSphere Application Server Network Deployment.

4.5 Sample team

The same sample team will be used to illustrate every pattern:

Barbara: The pricing analyst with overall responsibility for business performance, setting the business goals, and reacting to market changes. Barbara adopts the role of the Release Sponsor, setting the objectives for any changes, and approving the deployment of any release into production.

Paul: The pricing manager is responsible for managing the Pricing policies team and coordinating the various activities that are needed in the release. He adopts the role of Release Manager.

Rachel: A regional manager with responsibility for managing the pricing policies for New Jersey. Rachel is a release contributor in these scenarios, and she applies her experience to authoring rules in specific areas of the pricing policies.

Adam: The ODM administrator with responsibility for testing and deploying decisions. In these scenarios, Adam is a Release Contributor, but also undertakes a role liaising with the Configuration Management solutions to ensure that all deployments to production systems are compliant with the insurance organization's operations management policies.

Ivan: The IT Architect has responsibility for a team of developers that developed the Web Quoting application and its integration with IBM Operational Decision Manager. Ivan has responsibility for any Technical change and validation activities. Ivan represents the IT Architect, Rule Developer, and Integration Developer roles.

Terry: The QA Engineer has responsibility for system integration testing (SIT) and user acceptance testing (UAT) environments and ensuring that the quality objectives for the solution are maintained.

4.6 Simple pattern

This pattern is aimed at small teams that manage all the rules of an organization.

In this pattern, the full set of roles defined in Chapter 3, "Roles and responsibilities in governing decisions" on page 21 is not needed, and decision governance is covered by just two Decision Center roles with coarse-grained responsibilities:

- ▶ Administrators have full control of the Decision Center initializing the repository and managing deployments.
- ▶ Users have full control of every Decision Center artifact from the Business Console. They can fully manage decision governance framework releases, activities, and decision service artifacts.

This pattern only relies on the predefined Decision Center UI roles described in "The Predefined Decision Center UI roles" on page 37, and implements the basic security necessary to allow authenticated users to access the Business Console.

The roles and responsibilities as described in Chapter 3, "Roles and responsibilities in governing decisions" on page 21, are each associated to one of the two groups in Figure 4-3 on page 37.

Roles & Responsibilities		Decision Center Roles	
		Administrators	Users
Primary Business Roles	Release Manager	X	
	Business User		X
	Business Policy Analyst		X
	Rule Author		X
	Rule Tester		X
	Rule Validator		X
	Business Analyst		X
Primary Technical Roles	Rule Developer		X
	ODM Decision Center	X	
	Permission Manager		X
	ODM Decision Server	X	
	Deployment Manager	X	
	Deployer		X
	ODM Architect		X
	IT Integration developer		X
Secondary Business Roles	IT Quality Insurance (QA)		X
	Business Owner		
	Change Control Board		
Secondary Technical Roles	Subject Matter Expert (SME)		
	Application Server Administrator		
	LDAP Administrator		
	Database Administrator		
	System Administrator		
	Network Administrator		
	IT Architect		

Figure 4-3 Simple pattern roles and Decision Center groups

To illustrate the simple security configuration pattern, the security configuration process performed by the Application Server Administrator, providing LDAP server is not involved, consists simply of the following simple tasks:

- ▶ Defining IBM WebSphere Application Server users and groups in the WebSphere Application Server Administrative Console (task 3)
- ▶ Mapping the WebSphere Application Server groups to the predefined Decision Center UI Roles (task 4)

4.6.1 The Predefined Decision Center UI roles

This section describes the four predefined Decision Center UI roles.

rtsUser (Standard User)

The rtsUser role permits access to Decision Center, both the Business Console and Enterprise Console. When project security is not enabled, this role allows full access to every artifact within every project.

rtsConfigManager (Configuration Manager)

The `rtsConfigManager` role has all the rights of the `rtsUser` and additionally manages the connection between Decision Center and the Decision Server. The role is used to configure the deployment of a decision service to one or more staging environments and then onto production.

rtsAdministrator (Administrator) and rtsInstaller (Installer)

The ODM administrator should have both `rtsAdministrator` and `rtsInstaller` roles:

- ▶ The `rtsAdministrator` role has all the rights of the `rtsUser` and `rtsConfigurator` roles and additionally gives full access to the Decision Center server configuration and Decision Center security and permission. It also bypasses decision service project access control.
- ▶ The `rtsInstaller` role gives the user access to the Installation Settings wizard in the Enterprise Console, and is typically used once after the Decision Center is first started to complete the Decision Center installation.

4.6.2 Configuring WebSphere Application Server users and groups

The Application Server Administrator is required to configure application server security. First, create the WebSphere Application Server users and groups for accessing Decision Center.

Creating users

From the WebSphere Application Server admin console (<http://.../admin>), go to the **Users and Groups** → **Manage Users** screen and create the ODM users. In the sample case, the user list is shown in Figure 4-4.

Create...	Delete	Select	Select an action...				
Select	User ID	First name	Last name	E-mail	Unique Name		
<input type="checkbox"/>	adam	Adam	The ODM Administrator		uid=adam,o=defaultWIMFileBasedRealm		
<input type="checkbox"/>	admin	admin	The WAS Admin		uid=admin,o=defaultWIMFileBasedRealm		
<input type="checkbox"/>	barbara	Barbara	The Pricing Analyst		uid=barbara,o=defaultWIMFileBasedRealm		
<input type="checkbox"/>	ivan	Ivan	The IT Architect		uid=ivan,o=defaultWIMFileBasedRealm		
<input type="checkbox"/>	paul	Paul	The Pricing Manager		uid=paul,o=defaultWIMFileBasedRealm		
<input type="checkbox"/>	rachel	Rachel	The Regional Manager		uid=rachel,o=defaultWIMFileBasedRealm		
<input type="checkbox"/>	terry	Terry	The QA Engineer		uid=terry,o=defaultWIMFileBasedRealm		

Figure 4-4 WebSphere Application Server Decision Center sample users

Creating groups

Go to **Users and Groups** → **Manage Groups** screen and create two ODM groups to correspond to the two Decision Center groups described in Simple pattern roles and Decision Center groups. In the sample case the two groups are called `dcAdministrators` and `dcUsers`, as shown in WebSphere Application Server Decision Center sample groups shown in Figure 4-5 on page 39.

Tip: We suggest that you avoid using groups named and mapped identically to the predefined roles. Doing so is error prone, because mixing the purpose of the predefined Decision Center UI roles with the concept of access control and permissions management is confusing.

Select	Group name	Description	Unique Name
<input type="checkbox"/>	dcAdministrators		cn=dcAdministrators,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	dcUsers		cn=dcUsers,o=defaultWIMFileBasedRealm

Figure 4-5 WebSphere Application Server Decision Center sample groups

In the sample case, Adam is the only member of the dcAdministrators group, (Figure 4-6).

Group name
dcAdministrators

The group has 1 members.

Add Users... **Add Groups...** **Remove**

Select	ID	Type	Unique Name
<input type="checkbox"/>	adam		uid=adam,o=defaultWIMFileBasedRealm

Figure 4-6 WebSphere Application Server dcAdministrators group

All other users, Barbara, Ivan, Paul, Terry and Rachel are members of the dcUsers group (Figure 4-7).

Group name
dcUsers

The group has 5 members.

Add Users... **Add Groups...** **Remove**

Select	ID	Type	Unique Name
<input type="checkbox"/>	barbara		uid=barbara,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	ivan		uid=ivan,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	paul		uid=paul,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	rachel		uid=rachel,o=defaultWIMFileBasedRealm
<input type="checkbox"/>	terry		uid=terry,o=defaultWIMFileBasedRealm

Figure 4-7 WebSphere Application Server dcUsers group

4.6.3 Configuring Decision Center groups to roles mapping

To associate the WebSphere Application Server groups to the predefined Decision Center UI roles, complete the following steps:

1. Go to **Application** → **Application Type** → **WebSphere Applications**.
2. Select **teamserver** application.
3. Select **Security role to user/group mapping**.

The resulting application server configuration should be as follows (Figure 4-8).

Select	Role	Special subjects	Mapped users	Mapped groups
<input type="checkbox"/>	rtsUser	None		dcUsers
<input type="checkbox"/>	rtsAdministrator	None		dcAdministrators
<input type="checkbox"/>	rtsConfigManager	None		dcAdministrators
<input type="checkbox"/>	rtsInstaller	None		dcAdministrators

Figure 4-8 Simple pattern Security role to group mapping

4.7 Fine-grained pattern

The Fine-grained pattern extends the Simple pattern by enabling fine-grained access and artifact permissions to the Decision Center users.

It consists of defining finer user groups with limited access to artifacts of a decision service depending on their role.

This requires project security to be enabled at the decision service level, allowing only selected user groups to access specific decision service artifacts.

To illustrate the Fine-grained security configuration pattern, the security configuration process, providing LDAP server is not involved, consists of the selected tasks from Table 4-1 on page 35, (shown in Table 4-2).

Table 4-2 *Fine-Grained security configuration pattern tasks*

Number	Admin	Tasks	Chapter sections
3	Application Server	Define application server users and groups	"Configuring WebSphere Application Server users and groups" on page 38
4		Configure Decision Center application 's group to role mapping	"Configuring Decision Center groups to roles mapping" on page 39
7	ODM	Manually create the Decision Center users and groups	"Creating Decision Center users and groups" on page 44
8		Enforce decision service project security	"Enforcing Decision Center project security" on page 46
9		Configure project permissions	"Configuring Decision Center permissions" on page 47

4.7.1 Decision Center group permissions

When Decision Center project security is enforced, access to decision service artifacts is controlled. Otherwise every user with the `rtslUser` role has full control on every decision service artifact (as in the Simple pattern). When project security is enabled for a decision service, the user must belong to a Decision Center group that permits access to the decision service and permits access to artifacts within that decision service. These groups are defined using the Business Console.

The following list describes some important information to know about permissions:

- ▶ When project security is enforced, the default permission is None.
- ▶ Access to a release is by default inherited from the parent release, but can also be specifically set.
- ▶ User groups are given access to a set of decision service artifacts (rules, ruleflow, operation, variable, and so on) with View, Create, Update, or Delete permissions. In some cases, this permission may be limited to artifacts with a given name.
- ▶ User permission is the combination of the permissions of all groups that the user belongs to. In case of conflict, the least permissive option is chosen.

Note: View, Create, Update, and Delete

Some readers might be familiar with create, retrieve, update, and delete permissions (CRUD). In ODM, the read permission is called *view*, so the acronym you will see becomes VCUD for view, create, update, and delete.

4.7.2 Decision Center roles permission requirements

When enabled, Decision Center project security manages access control to artifacts.

Tip: When not enabled, full control on all artifacts to all Decision Center users is entitled. This is *not* recommended.

The object of this chapter is to group together ODM Business and Technical user roles (Figure 4-9 on page 42) that participate in decision service management and have common permission requirements.

The following roles are decision service-specific:

- ▶ *Users* are all the users involved in a decision service and are granted View (Read) access to all the decision service artifacts and Update access to Releases and Activities because any of them can be the owner, approver, author, or tester of releases and change activities.
- ▶ *Release Managers* manage Releases, Change, and Validation Activities from creation to completion.
- ▶ *Rule Testers* are Validation Activity testers and must be able to fully control tests and simulations before completing the activity.
- ▶ *Rule Authors* are business users responsible for editing action rules and decision tables as well as setting up the tests and simulations necessary to validate them. They are Change Activity owners and must be able to update their status.
- ▶ *Rule Developers* are ODM IT users responsible for the development of all decision service artifacts. They initialize the decision service project and are responsible for the most complex change activities. They are Change Activity owners and must be able to update the change status.

ODM Permission Elements		Decision Center Roles				
		User	Release Manager	Rule Tester	Rule Author	Rule Developer
Governance	Release	VU	VCUD	V	V	V
	Change Activity	VU	VCUD	V	VU	VU
	Validation Activity	VU	VCUD	VU	V	V
Decision Service Business Artifacts	Action Rule	V	V	V	VCUD	VCUD
	Decision Table	V	V	V	VCUD	CVUD
	Rule Package	V	V	V	V	CVUD
	Variable Set	V	V	V	V	CVUD
	Resource	V	V	V	V	VCUD
	Rule Package	V	V	V	V	VCUD
	Operation	V	V	V	V	VCUD
	Query	V	V	V	V	VCUD
	Ruleflow	V	V	V	V	VCUD
	BOM	V	V	V	V	VCUD
Decision Service IT Artifacts	BOM2XOM	V	V	V	V	VCUD
	Vocabulary	V	V	V	V	VCUD
	Function	V	V	V	V	VCUD
Test & Validation	Test Suite	V	V	VCUD	VCUD	VCUD
	Simulation	V	V	VCUD	VCUD	VCUD
	Simulation Configuration	V	V	VCUD	VCUD	VCUD
	Input Data	V	V	VCUD	VCUD	VCUD
	KPI	V	V	VCUD	VCUD	VCUD
	Metric	V	V	VCUD	VCUD	VCUD
	Simulation Model	V	V	VCUD	VCUD	VCUD
Deploy	Deployment	V	V	V	V	V
Enterprise Console	Technical Rules	V	V	V	V	V
	Template	V	V	V	V	V
	Decision Tree	V	V	V	V	V
	Smart View	V	V	V	V	V
	Vocabulary	V	V	V	V	V

Figure 4-9 Fine-Grained Decision Center project artifact permissions

4.7.3 Decision Center Permissions Groups

The Decision Center roles shown in Figure 4-9 lead to Figure 4-10 on page 43, where these Decision Center roles have been split between Decision Center groups with common permission requirements which, when appropriately combined, correspond to the Decision Center roles:

- ▶ The *Decision Center Users* group includes all users allowed to access the decision service. They are entitled to view every artifact and update a decision governance artifact, release, and activities, to participate in the decision governance workflow.
- ▶ The *Decision Center Release Managers* are all users organizing the decision service implementation. and therefore entitled to create and delete decision governance artifacts.
- ▶ The *Decision Center Rule Testers* are entitled to Create, Update, and Delete test and Simulation to perform Validation Activities.
- ▶ The *Decision Center Rule Authors* are only entitled to Create, Update, and Delete Action Rules and Decision Tables.
- ▶ The *Decision Center Rule Developers* are entitled full control of every remaining decision service artifact, except the ones specific to Classic Rule Project and Enterprise Console.

		ODM Permission Elements	Decision Center Groups				
			User	Release Manager	Rule Tester	Rule Author	Rule Developer
Governance	Release		VU	CD			
	Change Activity		VU	CD			
	Validation Activity		VU	CD			
Decision Service Business Artifacts	Action Rule		V			CUD	
	Decision Table		V			CUD	
	Rule Package		V				CUD
	Variable Set		V				CUD
	Resource		V				CUD
	Rule Package		V				CUD
	Operation		V				CUD
	Query		V				CUD
	Ruleflow		V				CUD
Decision Service IT Artifacts	BOM		V				CUD
	BOM2XOM		V				CUD
	Vocabulary		V				CUD
	Function		V				CUD
Test & Validation	Test Suite		V		CUD		
	Simulation		V		CUD		
	Simulation Configuration		V		CUD		
	Input Data		V		CUD		
	KPI		V		CUD		
	Metric		V		CUD		
	Simulation Model		V		CUD		
Deploy	Deployment		V				
Enterprise Console	Technical Rules		V				
	Template		V				
	Decision Tree		V				
	Smart View		V				
	Vocabulary		V				

Figure 4-10 Fine-Grained Decision Center groups permissions

Figure 4-11 on page 44 associates each ODM decision governance framework role, that is, release and activity owner, tester, and author (Chapter 7, “Decision governance framework” on page 85) to the necessary Decision Center roles and therefore groups:

- ▶ Every decision governance role is part of the Decision Center Users group.
- ▶ Owner role is the only one allowed to create Releases and Activities so performs the Release Manager role from Figure 4-12 on page 44.
- ▶ Approver role only needs to have the User role from Figure 4-12 on page 44 to be able to update a Release.
- ▶ Tester role needs full control on Tests and Simulations and this is covered by the Rule Tester role from Figure 4-12 on page 44.
- ▶ Business Release Author role needs to be able to Create, Update and Delete Action Rules and Decision Tables as covered by the Rule Author role from Figure 4-2 on page 34, as well as perform the Tester role.
- ▶ Technical Release Author role requires full control on every decision artifact except the Releases and Activities so must be a member of all roles shown in Figure 4-12 on page 44 except Release Manager.

Decision Governance Framework Role		Decision Center Groups				
		User	Release Manager	Rule Tester	Rule Author	Rule Developer
Owner		X	X			
Approver		X				
Tester		X		X		
Author	Business release	X		X	X	
	Technical Release	X		X	X	X

Figure 4-11 Decision governance framework role mapping to Fine-Grained Decision Center groups

In the sample case, users and Decision Center groups are associated in the Figure 4-12.

Decision Governance Framework Role		Decision Center Groups				
		Users (dcUsers)	Release Manager (dcReleaseManagers)	Rule Tester (dcTesters)	Rule Author (dcAuthors)	Rule Developer (dcDevelopers)
Owner	(Paul)	X	X			
Tester	(Rachel)	X		X		
Author	Business release (Barbara)	X		X	X	
	Technical Release (Ivan)	X		X	X	X

Figure 4-12 Sample user role mapping to Fine-Grained Decision Center groups

4.7.4 Creating Decision Center users and groups

The Business Console allows the ODM administrator to create users and groups. The user IDs are passed on from the application server when users are authenticated, but the groups are specific to Decision Center. Manual User and Group management is simple and best suited to small teams:

1. From the Business Console **ADMINISTRATION** → **Users** tab, click the circled **Plus sign (+)** to create the users corresponding to the ones created for the predefined Decision Center UI roles described in 4.6.2, “Configuring WebSphere Application Server users and groups” on page 38 (Figure 4-13 on page 45).

- From the **ADMINISTRATION** → **Groups** tab, click the circled **Plus sign (+)** to create the necessary groups identified in section 4.7.3, “Decision Center Permissions Groups”, as shown in Figure 4-14.

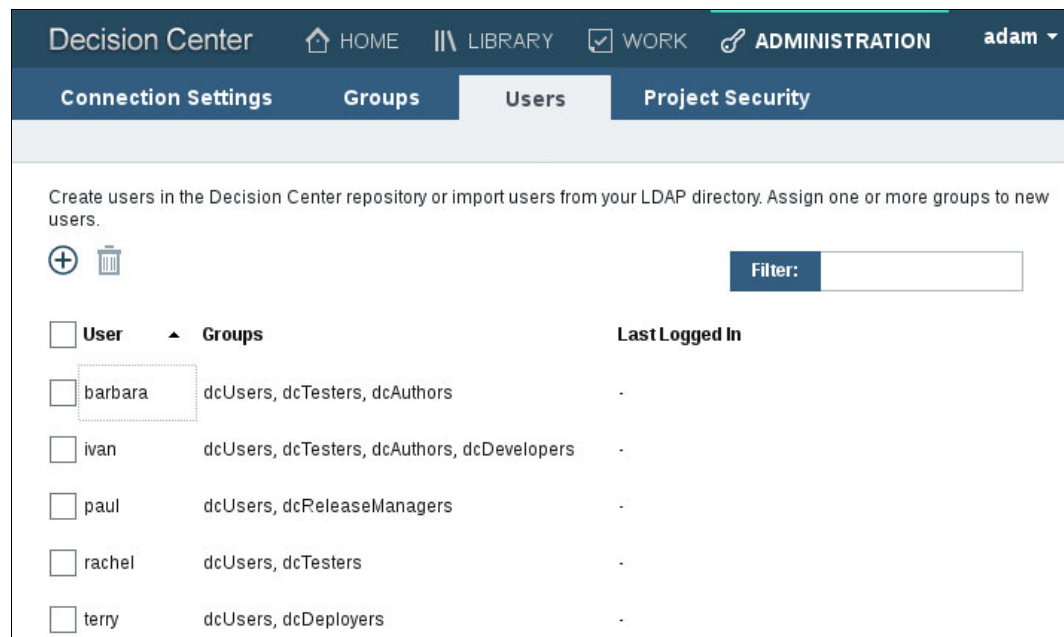


Figure 4-13 Fine-Grained Decision Center users

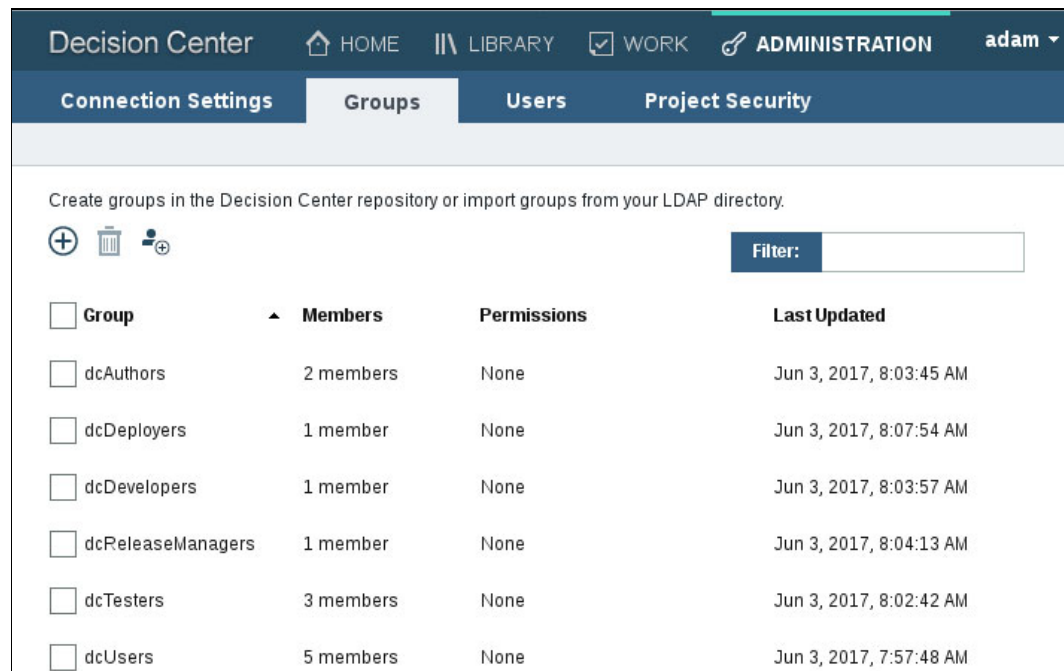


Figure 4-14 Fine-Grained Decision Center groups

4.7.5 Enforcing Decision Center project security

Decision service project security is enabled from the Business Console screen **ADMINISTRATION** → **Project Security** tab, as shown in Figure 4-15.

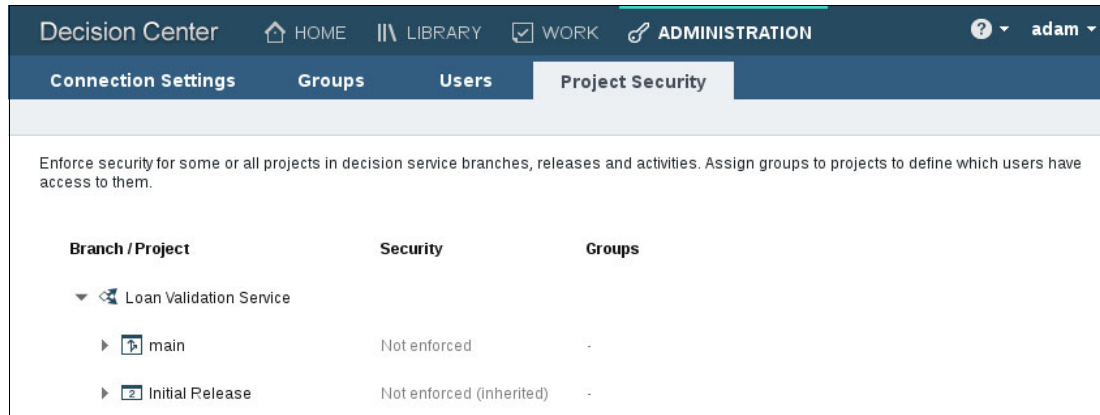


Figure 4-15 Loan Validation Service project security not enforced

To enforce Decision Center project security, complete the following steps:

1. From the top level of the branch hierarchy, hover the mouse next to the **Security** column header and click the **pen icon**, as shown in Figure 4-16.

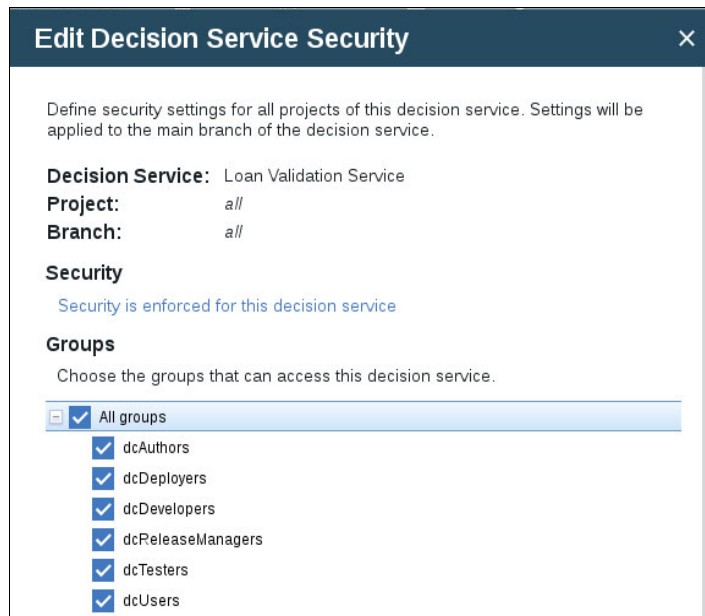


Figure 4-16 Loan Validation Service project security setting

2. Select the blue line, **Security is not enforced for this decision service**, beneath **Security** to switch to **Enforce security**. Select the LDAP or manually created groups to enforce access controls on the decision service, Figure 4-17.

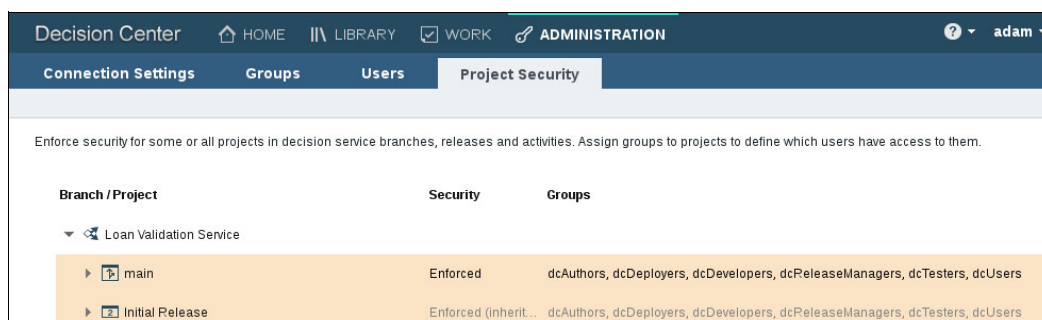


Figure 4-17 Loan Validation Service project security enforced

4.7.6 Configuring Decision Center permissions

The ODM administrator sets group permissions from the Enterprise Console via the **Configure** tab and **Security** → **Edit Permissions** screen:

1. Select a Group click **New**. It is now possible to add a new record to the group permissions list. Each permission record is made up of the following values:
 - The **Permission**: View, Create, Update or Delete.
 - The **Value** yes when permission is entitled or no otherwise.
 - The **Type** (such as artifact) to which the permission applies. Star (*) can be used when the permission applies to all of them. See the ODM Permission Element column on Table 4-13 on page 45 or Table 4-13 on page 45 for the available Type.
 - The **Property** can optionally be set with Update permission, to apply it to a given property of the Type, such as Name, documentation, Status, and so forth.
2. When permissions are set for every group, it is possible to verify that the resulting combinations of groups provide the expected permissions by using the **Configure** → **Security** → **View Effective Permissions** screen.

Per the information in Table 4-2 on page 40, group permissions are set as described in the following sections.

dcUsers group permissions

The Decision Center Users group is granted the following permissions, as shown in Figure 4-18:

- View to every artifact (*)
- Update to Release, Change Activity, and Validation Activity

You are currently editing the permissions for the group: **dcUsers**

Actions	PERMISSION	TYPE	PROPERTY	VALUE
<input type="checkbox"/>	View	*	-	Yes
<input type="checkbox"/>	Update	Release	*	Yes
<input type="checkbox"/>	Update	Change Activity	*	Yes
<input type="checkbox"/>	Update	Validation Activity	*	Yes

Figure 4-18 Fine-Grained dcUsers group permissions

As a reminder, all users have this permission configuration, essentially giving them the possibility to view every artifact in the decision service.

dcReleaseManagers group permissions

The Decision Center Release Managers group is granted Create and Delete permissions on Release, Change Activities, and Validation Activities, as shown in Figure 4-19.

You are currently editing the permissions for the group: dcReleaseManagers

Actions	PERMISSION	TYPE	PROPERTY	VALUE
<input type="checkbox"/>	▲ ▼ Create	Release	-	Yes
<input type="checkbox"/>	▲ ▼ Create	Change Activity	-	Yes
<input type="checkbox"/>	▲ ▼ Create	Validation Activity	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Release	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Change Activity	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Validation Activity	-	Yes

Figure 4-19 Fine-Grained dcReleaseManagers group permissions

dcTesters group permissions

The Decision Center Rule Testers group is granted Create, Update, and Delete permissions on Test and Simulations artifacts, as shown in Figure 4-20.

You are currently editing the permissions for the group: dcTesters

Actions	PERMISSION	TYPE	PROPERTY	VALUE
<input type="checkbox"/>	▲ ▼ Create	Test Suite (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Create	Simulation Configuration (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Create	Simulation Input Data (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Create	Simulation KPI (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Create	Simulation Metric (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Create	Simulation Report Format (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Update	Test Suite (Business Console)	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Simulation Configuration (Business Console)	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Simulation Input Data (Business Console)	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Simulation KPI (Business Console)	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Simulation Metric (Business Console)	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Simulation Report Format (Business Console)	*	Yes
<input type="checkbox"/>	▲ ▼ Delete	Test Suite (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Simulation Configuration (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Simulation Input Data (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Simulation KPI (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Simulation Metric (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Simulation Report Format (Business Console)	-	Yes

Figure 4-20 Fine-Grained dcTesters group permissions

dcAuthors group permissions

The Decision Center Rule Authors group is granted Create, Update, and Delete permissions on Action Rules and Decision Tables artifacts, as shown in Figure 4-21 on page 49.

You are currently editing the permissions for the group: dcAuthors

Actions	PERMISSION	TYPE	PROPERTY	VALUE
<input type="checkbox"/>	▲ ▼ Create	Action Rule	-	No
<input type="checkbox"/>	▲ ▼ Create	Decision Table	-	No
<input type="checkbox"/>	▲ ▼ Update	Action Rule	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Decision Table	*	Yes
<input type="checkbox"/>	▲ ▼ Delete	Action Rule	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Decision Table	-	Yes

Figure 4-21 Fine-Grained dcAuthors group permissions

dcDevelopers group permissions

The Decision Center Rule Developers group is granted Create, Update, and Delete permissions on every artifact not included in Rule Testers, Rule Authors, and Release Managers permissions, as shown in Figure 4-22.

You are currently editing the permissions for the group: dcDevelopers

Actions	PERMISSION	TYPE	PROPERTY	VALUE
<input type="checkbox"/>	▲ ▼ Create	Action Query	-	No
<input type="checkbox"/>	▲ ▼ Create	BOM	-	No
<input type="checkbox"/>	▲ ▼ Create	BOM to XOM Mapping	-	No
<input type="checkbox"/>	▲ ▼ Create	Decision Operation (Business Console)	-	No
<input type="checkbox"/>	▲ ▼ Create	Folder	-	No
<input type="checkbox"/>	▲ ▼ Create	Function	-	No
<input type="checkbox"/>	▲ ▼ Create	Resource	-	No
<input type="checkbox"/>	▲ ▼ Create	Ruleflow	-	No
<input type="checkbox"/>	▲ ▼ Create	Variable Set	-	No
<input type="checkbox"/>	▲ ▼ Create	Vocabulary	-	No
<input type="checkbox"/>	▲ ▼ Update	Action Query	*	Yes
<input type="checkbox"/>	▲ ▼ Update	BOM	*	Yes
<input type="checkbox"/>	▲ ▼ Update	BOM to XOM Mapping	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Decision Operation (Business Console)	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Folder	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Function	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Resource	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Ruleflow	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Variable Set	*	Yes
<input type="checkbox"/>	▲ ▼ Update	Vocabulary	*	Yes
<input type="checkbox"/>	▲ ▼ Delete	Action Query	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	BOM	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	BOM to XOM Mapping	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Decision Operation (Business Console)	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Folder	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Function	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Resource	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Ruleflow	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Variable Set	-	Yes
<input type="checkbox"/>	▲ ▼ Delete	Vocabulary	-	Yes

Figure 4-22 Fine-Grained dcDevelopers group permissions

4.8 Project pattern

The project pattern now extends the Fine-grained pattern:

- ▶ Adding a decision service-specific group to replace the dcUsers group in project permissions configuration to filter Decision Center users who are allowed to access the decision service.
- ▶ Managing users and groups using an LDAP server
- ▶ Singularizing the deployment manager from the administrator to delegate this responsibility to more operational teams.

To illustrate the Project security configuration pattern, the security configuration process performed by the LDAP, Application Server, and ODM Decision Center Administrators consists of the selected tasks from Table 4-1 on page 35, shown in Table 4-3.

Table 4-3 Project security configuration pattern tasks

Number	Admin	Tasks	Chapter sections
1	LDAP	Define user groups	4.8.1, "Defining LDAP Users & Groups" on page 53
2	Application Server	Configure security to rely LDAP	4.8.2, "Adding LDAP registry to WebSphere Application Server Global Security" on page 53
4		Configure Decision Center application's group to role mapping	4.8.3, "Configuring Decision Center role-to-group mapping " on page 53
5	ODM	Define LDAP connection	4.8.4, "Configuring a Decision Center LDAP connection" on page 54
6		Pull LDAP users and groups	4.8.5, "Uploading LDAP Users and Groups in Decision Center" on page 55
8		Enforce decision service project security	4.8.6, "Configuring Decision Center project security" on page 57
9		Configure project permission	4.7.6, "Configuring Decision Center permissions" on page 47; 4.8.7, "Configuring Decision Center permissions" on page 58

This last pattern demonstrates that after the LDAP configuration and the common Decision Center role-related groups are created, when creating a new decision service, only the decision service users group needs to be created. This action is followed by project permission configuration, as described in 4.7.6, "Configuring Decision Center permissions" on page 47 and 4.8.7, "Configuring Decision Center permissions" on page 58 (tasks 6 or 7 and 9 in Table 4-1 on page 35, shown in Table 4-4).

Table 4-4 New decision service security configuration tasks

Number	Admin	Task	Chapter section
6	ODM	Pull LDAP users and groups	4.8.5, "Uploading LDAP Users and Groups in Decision Center" on page 55
7		Manually define Decision Center users and groups	4.7.4, "Creating Decision Center users and groups" on page 44
9		Configure project permissions	4.7.6, "Configuring Decision Center permissions" on page 47 and 4.8.7, "Configuring Decision Center permissions" on page 58

Configuration described in the following sections extends the Simple and Fine-Grained patterns with the predefined Decision Center UI roles now following Figure 4-23 that gives the deployment manager the rtsConfigManager Decision Center UI role (this is described in further detail in Chapter 8, "Deployment" on page 119).

Roles & Responsabilites		Decision Center UI Roles			
		rtsAdminsitrator	rtsInstaller	rtsConfigManager	rtsUser
Primary Business Roles	Release Manager				X
	Business User				X
	Business Policy Analyst				X
	Rule Author				X
	Rule Tester				X
	Rule Validator				X
Primary Technical Roles	Business Analyst				X
	Rule Developer				X
	ODM Decision Center Administrator	X	X		
	Permission Manager				X
	ODM Decision Server Administrator				X
	Deployment Manager			X	
	Deployer				X
	ODM Architect				X
Secondary Business Roles	IT Integration developer				X
	IT Quality Insurance (QA)				X
	Business Owner				
Secondary Technical Roles	Change Control Board				
	Subject Matter Expert (SME)				
	Application Server Administrator				
	LDAP Adminsitrator				
	Database Adminsitrator				
	System Administrator				
	Network Adminsitrator				
	IT Architect				

Figure 4-23 Project pattern predefined Decision Center UI roles

The deployment manager role is then given full control of deployment configuration from the Business Console, and this modifies the ODM permission element configuration table, as shown in Figure 4-24.

		ODM Permission Elements	Decision Center Groups				
			User	Release Manager	Rule Tester	Rule Author	Rule Developer
Governance	Release	VU	CD				
	Change Activity	VU	CD				
	Validation Activity	VU	CD				
Decision Service Business Artifacts	Action Rule	V			CUD		
	Decision Table	V			CUD		
	Rule Package	V				CUD	
	Variable Set	V				CUD	
	Resource	V				CUD	
	Rule Package	V				CUD	
	Operation	V				CUD	
	Query	V				CUD	
Decision Service IT Artifacts	Ruleflow	V				CUD	
	BOM	V				CUD	
	BOM2XOM	V				CUD	
	Vocabulary	V				CUD	
	Function	V				CUD	
Test & Validation	Test Suite	V		CUD			
	Simulation	V		CUD			
	Simulation Configuration	V		CUD			
	Input Data	V		CUD			
	KPI	V		CUD			
	Metric	V		CUD			
	Simulation Model	V		CUD			
Deploy	Deployment	V					CUD
Enterprise Console	Technical Rules	V					
	Template	V					
	Decision Tree	V					
	Smart View	V					
	Vocabulary	V					

Figure 4-24 Project pattern Decision Center groups permissions

In the sample case, because a dedicated user group is entitled to work on the Loan Validation Service, the sample user role mapping to Decision Center groups is modified as shown in Figure 4-25.

Decision Governance Framework Role		Decision Center Groups					
		Users (lvsUsers)	Release Managers (dcReleaseManagers)	Rule Testers (dcRuleTesters)	Rule Authors (dcRuleAuthors)	Rule Developers (dcRuleDevelopers)	Deployment Managers (dcDeploymentManager)
Owner		X	X				
Approver		X					
Tester		X		X			
Author	Business release	X		X	X		
	Technical Release	X		X	X	X	
Deployment		X					X

Figure 4-25 Decision governance framework role mapping to Project pattern Decision Center groups

4.8.1 Defining LDAP Users & Groups

This task consists of configuring the same users and groups as in the Fine-grained pattern (see 4.7.4, “Creating Decision Center users and groups” on page 44) but from the LDAP registry administration application, which is dependent on LDAP server technology.

4.8.2 Adding LDAP registry to WebSphere Application Server Global Security

By default, WebSphere Application Server user account repository is based on the Federated Repository aggregating several user repositories into one from the perspective of the WebSphere Application Server.

In this sample case, the LDAP registry is added from the WebSphere Application Server admin console:

1. Go to the **Security** → **Global security** screen.
2. Click **Configure**.

The resulting federated repository should be similar to Figure 4-26.

Select	Base Entry	Repository Identifier	Repository Type
You can administer the following resources:			
<input type="checkbox"/>	dc=odmredbooksample,dc=com	LDAP1	LDAP:CUSTOM
<input type="checkbox"/>	o=defaultWIMFileBasedRealm	InternalFileRepository	File

Figure 4-26 Project pattern WebSphere Application Server user account repository

4.8.3 Configuring Decision Center role-to-group mapping

As in 4.6.3, “Configuring Decision Center groups to roles mapping” on page 39, Decision Center UI roles are associated to groups. Here, in the Project security configuration pattern:

- The groups are the previously created LDAP groups.
- The dcDeploymentManager group is now given the rtsConfigManager role.

The resulting application server configuration should be as shown in Figure 4-27.

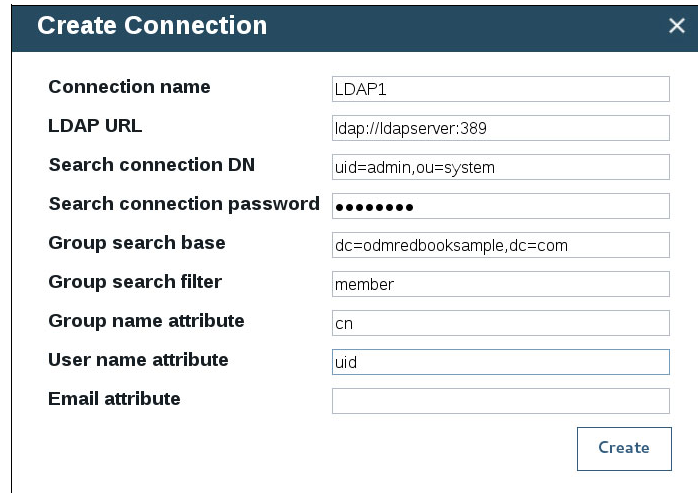
Select	Role	Special subjects	Mapped users	Mapped groups
<input type="checkbox"/>	rtsUser	None		dcUsers
<input type="checkbox"/>	rtsAdministrator	None		dcAdministrators
<input type="checkbox"/>	rtsConfigManager	None		dcDeploymentManager
<input type="checkbox"/>	rtsInstaller	None		dcAdministrators

Figure 4-27 Project pattern Decision Center role-to-group mapping

4.8.4 Configuring a Decision Center LDAP connection

To configure the LDAP connection, complete the following steps:

1. As ODM administrator, go to the **ADMINISTRATION** → **Connection Settings** screen.
2. Create an LDAP connection by clicking the circled **Plus sign (+)**. This takes you to the Create Connection screen shown in Figure 4-28.



Create Connection	
Connection name	LDAP1
LDAP URL	ldap://ldapsrvr:389
Search connection DN	uid=admin,ou=system
Search connection password	••••••••
Group search base	dc=odmredbooksample,dc=com
Group search filter	member
Group name attribute	cn
User name attribute	uid
Email attribute	
<button>Create</button>	

Figure 4-28 Create Decision Center LDAP connection

Figure 4-28 shows the following information:

- ▶ **Connection name** is the name of the connection.
- ▶ **LDAP URL** is the URL of the LDAP server to connect to.
- ▶ **Search connection DN** is used when binding to LDAP for searches. If not specified, an anonymous connection is made.
- ▶ **Search connection password** is the **Search connection DN** password.
- ▶ **Group search base** is the root path from which searches are made in the directory.
- ▶ **Group search filter** is the attribute that represents a member in a group.
- ▶ **Group name attribute** is the name of the entry that contains the user group names in the directory entries that are returned by the group search.
- ▶ **User name attribute** is the attribute that contains the user name in the LDAP.
- ▶ **Email attribute** is the attribute that contains the email address in the LDAP.

When created, the LDAP Connection appears in the list with status Enabled if it could successfully establish connection with the provided parameters, as shown in Figure 4-29.

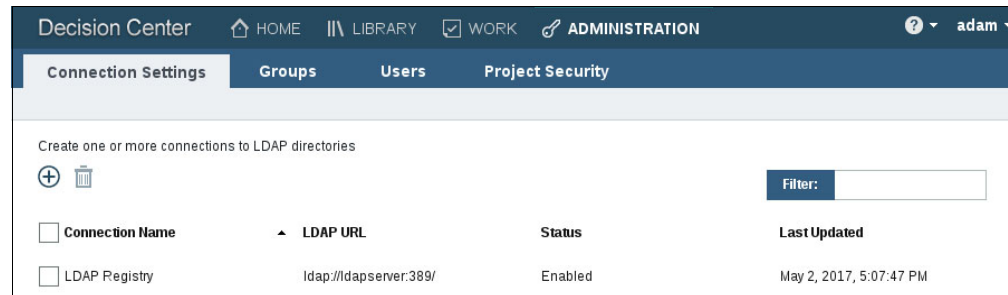


Figure 4-29 Resulting Decision Center LDAP connection setting

Tip: It is sometimes necessary to create several connections to the same LDAP registry because the LDAP Connection Setting parameters are limited to a single possible value and do not support complex LDAP filter expressions with special characters like parentheses, ampersand, and equal sign.

4.8.5 Uploading LDAP Users and Groups in Decision Center

After the necessary LDAP connections are created, complete the following steps:

1. Go to the **Administration** → **Groups** screen.
2. Click the **Down arrow** to display the result of the LDAP search, and select the necessary groups involved in decision service security, as shown in Figure 4-30.

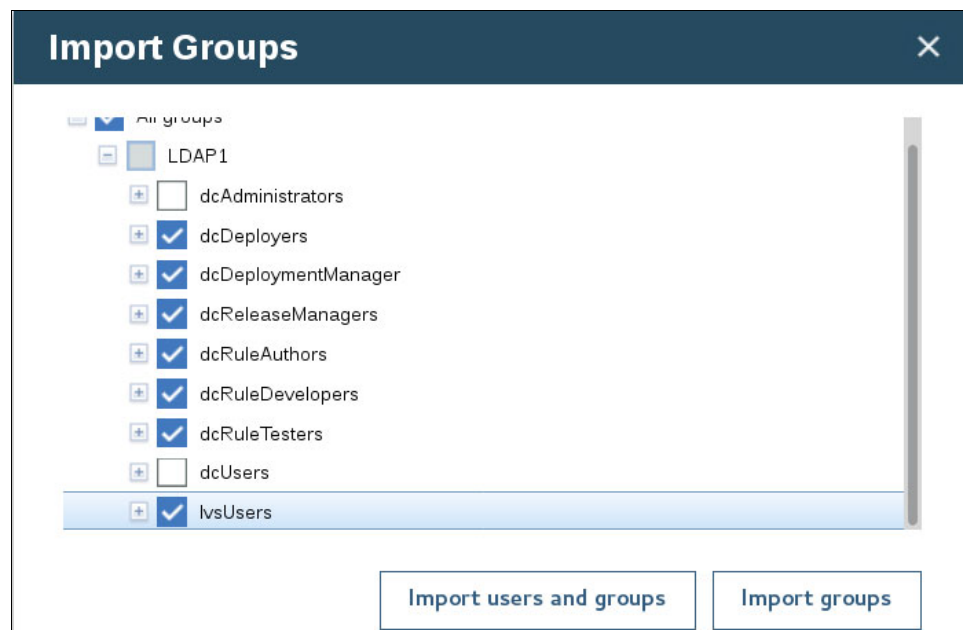


Figure 4-30 Importing groups from Decision Center LDAP registry

The group `dcDeployers`, different from the group `dcDeployerManager` identifies users given the responsibility to perform deployments. This is set by the deployment manager when configuring the deployment configuration (see Chapter 8, “Deployment” on page 119 for more details).

Tip: If the available LDAP groups do not fully fit the required Decision Center groups, manual groups can be created and populated with users recovered from the LDAP registries.

The resulting group list is shown in Figure 4-31.

Connection Settings	Groups	Users	Project Security
Create groups in the Decision Center repository or import groups from your LDAP directory.			
<input type="checkbox"/>	Group	▲	Members
<input type="checkbox"/>	dcDeployers		1 member
<input type="checkbox"/>	dcDeploymentManager		1 member
<input type="checkbox"/>	dcReleaseManagers		1 member
<input type="checkbox"/>	dcRuleAuthors		2 members
<input type="checkbox"/>	dcRuleDevelopers		1 member
<input type="checkbox"/>	dcRuleTesters		3 members
<input type="checkbox"/>	lvsUsers		5 members

Figure 4-31 Decision Center LDAP groups

The resulting user list in screen **Administration / Users** is shown in Figure 4-32.

Connection Settings	Groups	Users	Project Security
Create users in the Decision Center repository or import users from your LDAP directory. Assign one			
<input type="checkbox"/>	User	▲	Groups
<input type="checkbox"/>	barbara		dcRuleAuthors, dcRuleTesters, lvsUsers
<input type="checkbox"/>	ivan		dcRuleAuthors, dcRuleDevelopers, dcRuleTesters, lvsUsers
<input type="checkbox"/>	paul		dcReleaseManagers, lvsUsers
<input type="checkbox"/>	rachel		dcRuleTesters, lvsUsers
<input type="checkbox"/>	terry		dcDeployers, dcDeploymentManager, lvsUsers

Figure 4-32 Decision Center LDAP users

Tip: User and group lists are not dynamically updated in Decision Center and whenever a new user is added to the LDAP registry, the ODM administrator must update the Decision Center user list from the Business Console.

4.8.6 Configuring Decision Center project security

To configure project security, complete the following steps:

1. Decision Center project security is enabled from the Business Console screen. Go to **ADMINISTRATION** → **Project Security**, as shown in Figure 4-33.

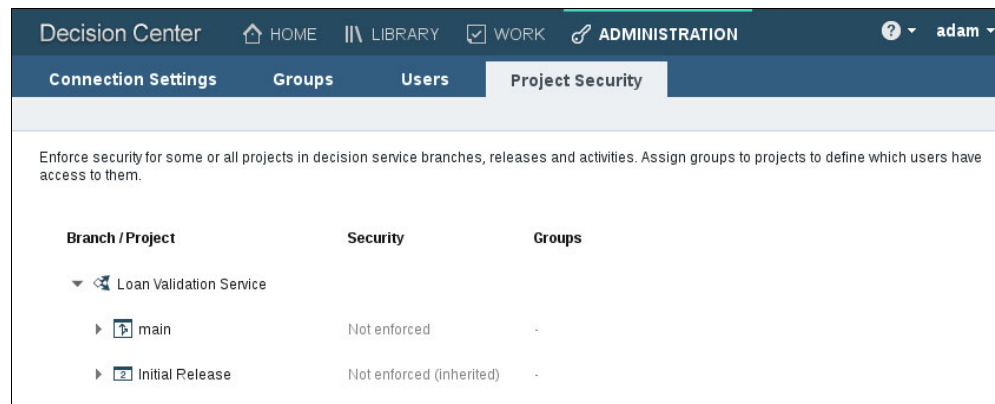


Figure 4-33 Project pattern Decision Center project Security not enforced

2. From the top level of the branch hierarchy, hover the mouse next to the **Security** column header and click the **Pen icon**, as shown in Figure 4-34.

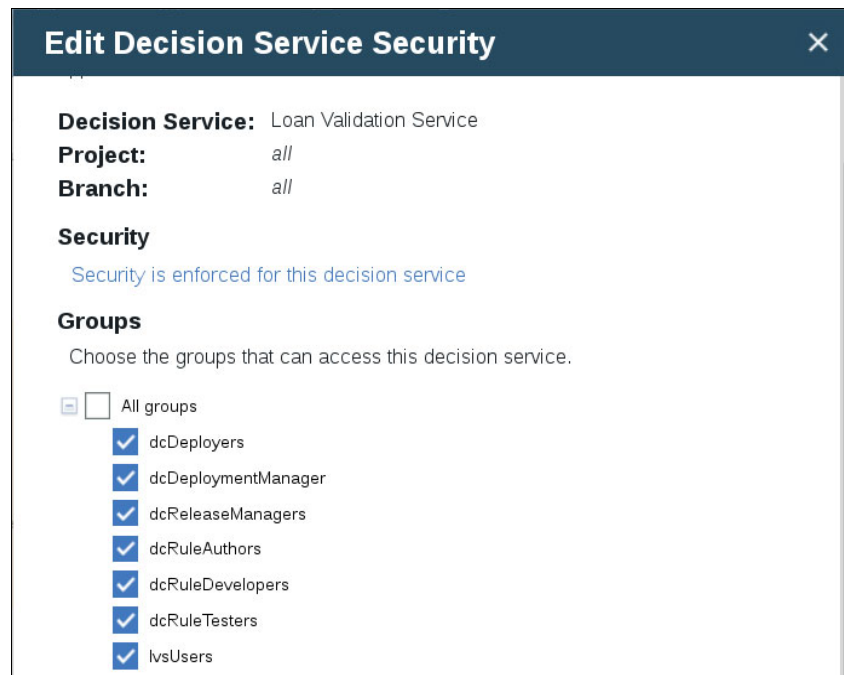


Figure 4-34 Project pattern Decision Center project Security setting

3. Select the blue line, **Security is not enforced for this decision service**, beneath **Security** to switch to **Enforce security** and select the LDAP groups that are granted access to the decision service. The resulting project security is shown in Figure 4-35.

Connection Settings	Groups	Users	Project Security
Enforce security for some or all projects in decision service branches, releases and activities. Assign groups to projects to define which users have access to them.			
Branch / Project	Security	Groups	
▼ Loan Validation Service			
▶ main	Enforced	dcDeployers, dcDeploymentManager, dcReleaseMan	

Figure 4-35 Project pattern Decision Center project Security enforced

4.8.7 Configuring Decision Center permissions

As in section 4.7.6, “Configuring Decision Center permissions” on page 47, group permission must be set following “Project pattern Decision Center groups permissions” on page 52. Adding groups `lvsUsers` and `dcDeploymentManager`.

The group `lvsUsers` are the users specifically allowed to view and work on the Loan Validation Service.

If a second decision service project, like Loan Underwriting Service, was added to the Decision Center, a group name, for instance `lusUsers`, would give the appropriate users access to this decision service.

`lvsUsers` group permissions

The Loan Validation Service Users group is granted the same permission as the Decision Center Users group, as shown in Figure 4-36.

You are currently editing the permissions for the group: <code>lvsUsers</code>					
Actions	PERMISSION	TYPE	PROPERTY	VALUE	
<input type="checkbox"/>	View	*	-	Yes	
<input type="checkbox"/>	Update	Release	*	Yes	
<input type="checkbox"/>	Update	Change Activity	*	Yes	
<input type="checkbox"/>	Update	Validation Activity	*	Yes	

Figure 4-36 Project pattern `lvsUsers` group permissions

dcDeploymentManager group permissions

The deployment managers group is granted Create, Update, and Delete permissions for Deployment Configurations, as shown in Figure 4-37.

You are currently editing the permissions for the group: dcDeploymentManager ▾










Actions	PERMISSION TYPE	PROPERTY VALUE
	  Create	Deployment Configuration (Business Console) - Yes
	  Update	Deployment Configuration (Business Console) * Yes
	  Delete	Deployment Configuration (Business Console) - Yes

Figure 4-37 Project pattern dcDeploymentManager group permissions

4.9 Conclusion

We showed that creating effective Decision Center user permissions is a combination of the following activities:

- ▶ Decision Center Application Server security configuration
- ▶ Decision service release and activity roles and status
- ▶ Decision service user group permissions

This chapter provided a suggested security and permissions configuration for your decision services, where decision governance roles listed in Chapter 3, “Roles and responsibilities in governing decisions” on page 21 are mapped to Decision Center Application Server roles, ODM decision governance framework roles, and Decision Center groups in the case of three patterns fitting most possible levels of organizational complexity.



Designing decision services

This chapter describes the goal of *operational decision management* to enable you to represent and manage high-value business decisions, and to make them executable in the context of an IT-based business solution. This decision-making capability can be considered a *service* in the context of a service-oriented architecture (SOA).

The IBM Operational Decision Manager (ODM) platform provides many artifacts to support the definition and the structuring of decision services, both in the context of the rule authoring phase (by using rule projects and rule packages artifacts) and the rules deployment phase (by using artifacts such as rulesets and RuleApps). This chapter describes how to use these constructs to structure a decision service.

This chapter covers the following topics:

- ▶ Concepts
- ▶ Project hierarchies
- ▶ Folder structure
- ▶ Orchestrating ruleset execution
- ▶ Advanced topics
- ▶ Conclusion

5.1 Concepts

This section describes some of the essential concepts used when discussing the organization of decision management.

5.1.1 Decision service

A *decision service* is a collection of rule projects that contain business rule artifacts (such as action rules). Within each rule project are rule packages that group rule artifacts in a functional hierarchy. Rule flows orchestrate the execution order of rule packages to determine the execution sequence of rule artifacts in response to a decision request.

IBM Operational Decision Manager groups rule artifacts, flows, and parameters in an executable unit called a ruleset. This ruleset has a defined signature in terms of inputs and outputs. The ruleset signature is exposed as a service directly via Java invocation or via SOAP/REST. The service can be used as the interface for testing activities.

A decision service is the unit of deployment of decisions in ODM. Business solutions call decision operations as a service. The decision service is managed, delivered, and deployed by decision governance.

The decision service contains the following components (Figure 5-1):

Deployment configuration	One or more destinations of the servers to which the service is deployed.
Decision operations	Defines one or more signatures for calling the decision service.
Business logic	The rules implementing the services.
Business object model	The vocabulary on which the business logic is based.

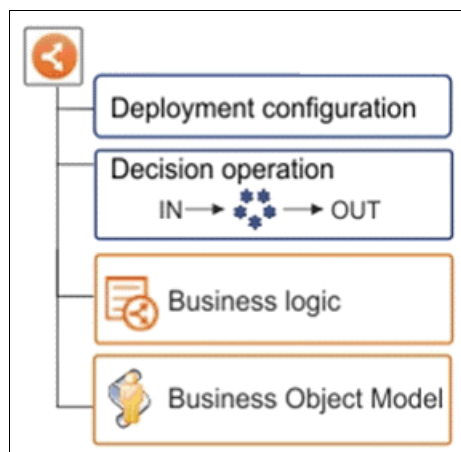


Figure 5-1 Decision service

The deployment manager manages the deployment configuration for the decision service. Each deployment configuration lists the associated decision operations and target servers, and defines the version policy for numbering the operations. A business solution calls a decision by using the name and version number of a decision operation.

The business user works with decision services in the Business Console. The Business Console provides governance services for managing, testing, and deploying decision services. Changes to the rules in a decision service are coordinated within a business release.

Decision operations

The *decision operation* defines which rules from a given branch are part of the ruleset. A decision operation includes all of the settings needed to define the contents of a ruleset and its parameters. You choose which decision operation to use when creating a test suite, simulation, or deployment configuration.

Client applications interact with a ruleset by using input and output parameters. These parameters are defined in the signature of a decision operation.

The parameters of a ruleset can have three directions:

- ▶ **Input:** The parameter value is provided as input to the ruleset on execution.
- ▶ **Output:** The parameter value is set by the execution of the ruleset and provided as output from the ruleset at execution completion.
- ▶ **Input/output:** The parameter value is provided as input to the ruleset on execution and its value can be modified by the ruleset and provided as output at execution completion.

You create the parameters from any of the ruleset variables that are available in the projects that are part of the scope of the decision operation. Ruleset variables are internal to a ruleset and provide a way to exchange data between rules, functions, and tasks.

Ruleset content

In the decision operation, the ruleset content is defined by specifying the following information:

- ▶ **The source rule project.** All the rules and variables contained in this project and any of its dependent projects become eligible to be included in the ruleset. This is usually the name of the decision service project.
- ▶ **Ruleflow.** The ruleflow to include in the ruleset and which is used for the orchestration of the execution of the rules.

Advanced configuration settings:

- ▶ **A query to filter the rules** so that only a subset of the rules of the source rule project and its dependent projects are included in the ruleset. Typically, this is used to include rules from specific folders of a project or according to a rule property such as status. You write queries in the Queries tab of the Business Console.
- ▶ **A validator to filter the rules.** A validator can be written in Rule Designer to select the required rules. If you specify both, the query is processed first, then the validation is applied.

5.1.2 Execution object model

The execution object model (XOM) (Figure 5-2) is the model against which you run rules. It references the application objects and data, and is the base implementation of the business object model (BOM). BOM Rule projects reference the XOM.

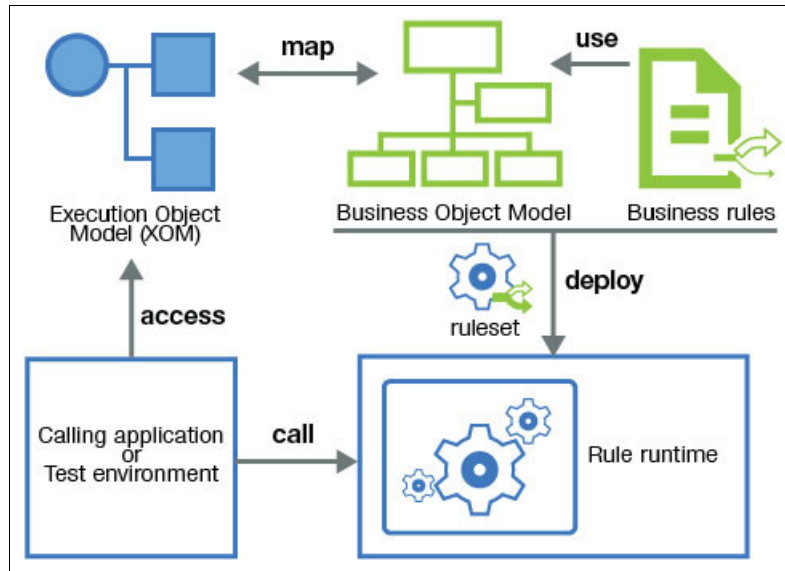


Figure 5-2 Executable object model

5.1.3 Business object model

You use the business object model (BOM) to make business rule editing easier by providing tools to set up a natural language vocabulary. With this vocabulary, policy managers can describe their business logic in a business rule language.

The BOM is the basis for the vocabulary used in business rules. It is an object model similar to a Java object model, and contains elements that map to those of the XOM.

BOM-to-XOM mapping defines the correspondence between the BOM and the XOM used at run time.

5.2 Project hierarchies

To offer the best flexibility over time and to make it easier to maintain and test rules, configure decision services in a rule project hierarchy. Projects can then be configured to refer to one another.

The project hierarchy should be used for the following functions:

- ▶ Separating decision service projects and decision operations from rule projects
- ▶ Grouping business rules by subject area, region, or another grouping that makes business sense to reduce the overall number of rules in a single project
- ▶ Separating BOM from business rules to allow BOM reuse among multiple rule projects
- ▶ Separating BOM entries by subject area to limit the size of the BOM to avoid potential performance issues

For example, if a client application calls different decision points, you should arrange the project structure to organize each decision operation and its ruleflow in a project. In addition, organize its unique rules in a separate rule project for that specific task. Furthermore, you can keep the BOM required for this specific set of rules separate to specify the vocabulary that is available only to those rules. This chapter describes some common project hierarchies.

The BOM is an entity that is reusable across the different operations of a service, and sometimes part of it reusable across the overall organization. It is also an entity that evolves less frequently than the rules during the decision lifecycle. Thus, it often makes sense to define a common BOM project for the decision service.

As a convention, no other artifact than the BOM is defined by this project. Any decision service involves at least one such BOM project, but there can be more than one depending on the complexity of the BOM and the scope of the decisions. These BOM projects are referenced (directly or transitively) as needed by all other rule projects to support the definition of the rule artifacts.

In Decision Center, releases apply to all of the projects in the hierarchy. Organizing an efficient project hierarchy enables you to manage rule change and deployment per specific project.

5.2.1 Diamond-shaped project hierarchy

In a diamond-shaped rule project organization, the decision service project contains a ruleflow that uses rules from other rule projects in the hierarchy. The rule projects share a common BOM that is kept in another rule project (Figure 5-3).

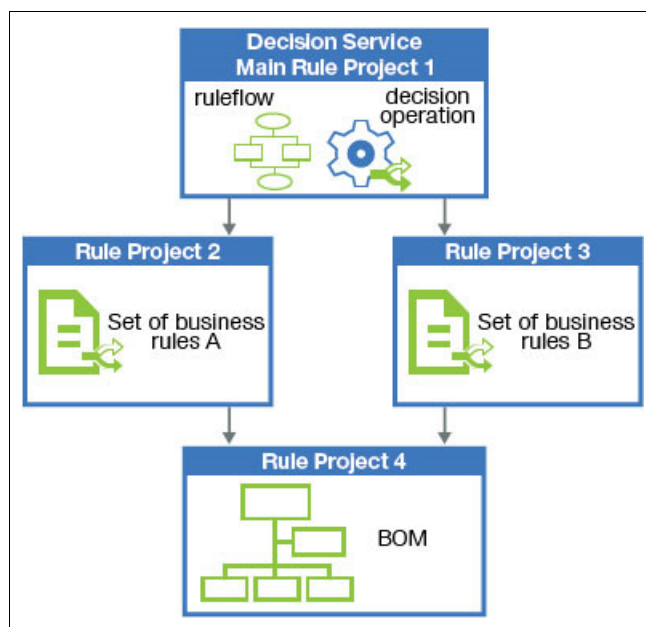


Figure 5-3 Diamond project structure hierarchy

5.2.2 Our sample project hierarchy evolution

Important: It is always easier to perform the restructuring of projects in Rule Designer before having synchronized with a Decision Center database. Doing so after synchronization requires careful planning and the execution of specific steps to avoid issues later.

To demonstrate how a project can evolve over time, we walk through the evolution of the project hierarchy for the sample decision service that we use throughout this IBM Redbooks publication.

We first start with one decision service and a single operation in that decision service, this service uses the rules in the Loan Validation Check project (Figure 5-4).

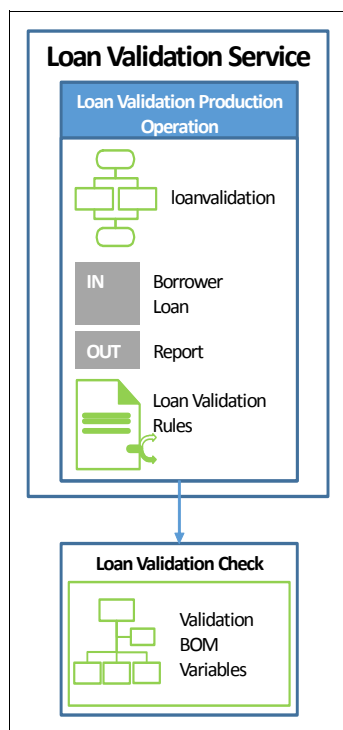


Figure 5-4 Loan Validation Service organization

Over time we might want to add new projects that share the same BOM and variables. To achieve this, we need to restructure the Loan Validation Check project and to separate the two main artifacts that it contains: The rules and the BOM.

Complete the following steps:

1. In Rule Designer, we create a Loan Validation Base project and move the BOM and variables to this new project and add the dependency from Loan Validation Check to Loan Validation Base.
2. We can then also create the two new projects required for organizing the rules: Loan Validation Determination and Loan Validation Scoring (Figure 5-5).

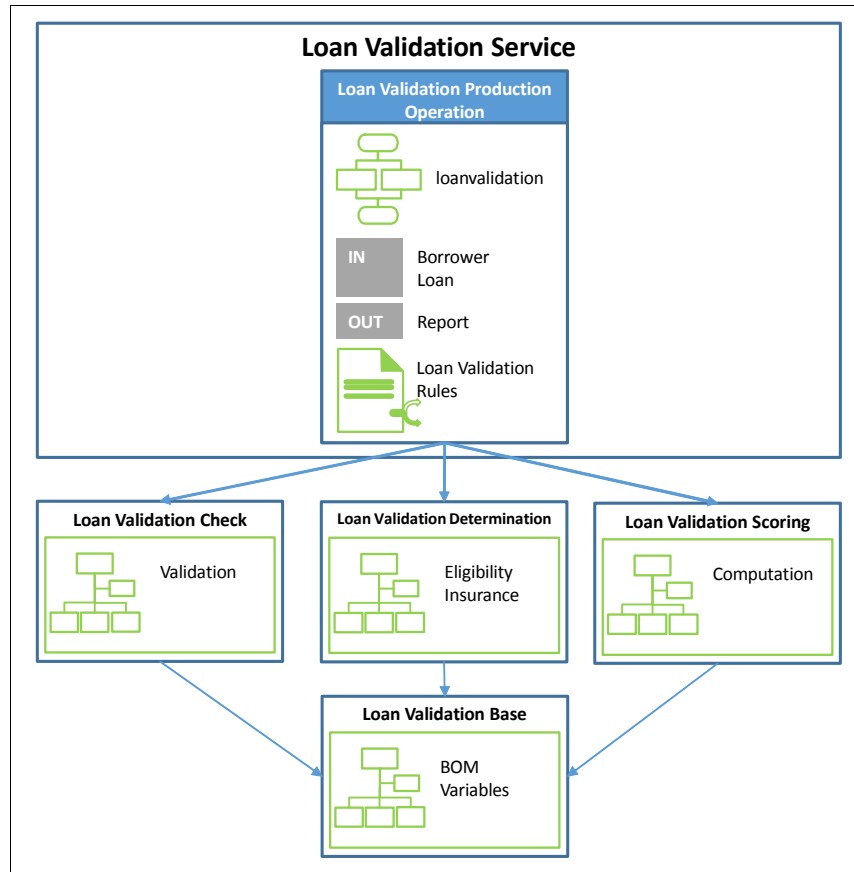


Figure 5-5 Loan Validation Service Organization - Refactored

- The final step of our reorganization requires the addition of a second operation that has additional return values in the response of the call to the service: The score and the grade (Figure 5-6).

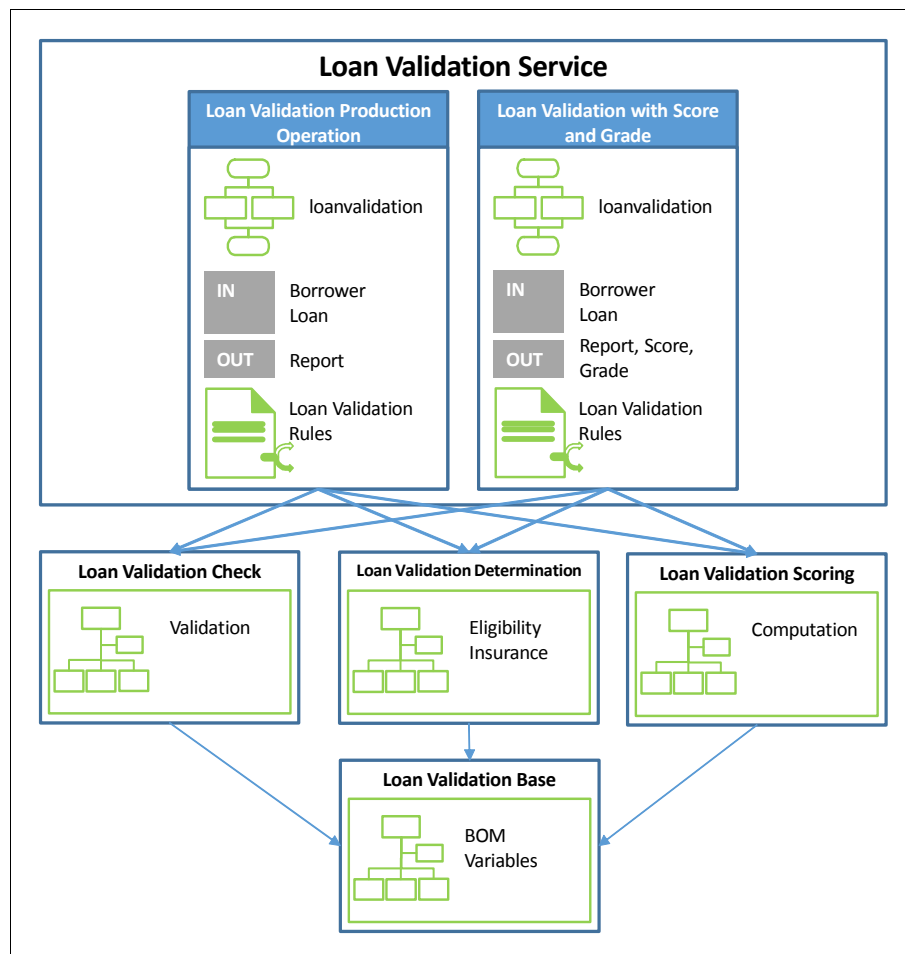


Figure 5-6 Loan Validation Service Organization - Final

Figure 5-6 shows the rule project structure for the Loan Validation Service.

The Service is separated into two operations:

- ▶ Loan Validation Production Operation
- ▶ Loan Validation with Score and Grade

The two services are virtually identical except that the second one returns two more parameters that contain the score and the grade for the loan application. They run the same ruleflow (loanvalidation).

5.3 Folder structure

To make it easier for rule authors to quickly and intuitively navigate to an existing rule or to a location where they can create a new one, rules are organized using a rule packages hierarchy. The packages hierarchy allows you to group rules that are logically related by a business dimension.

For example, the folder hierarchy follows the same steps as the decision process, and each package ends up being associated with rule tasks in the ruleflow. Subpackages can then be divided along other business dimensions, such as a specific product characteristic or the geography in which the rules apply.

As an example of the structure of the folders by business dimension, Figure 5-7 shows that the Loan Validation Check project contains a validation folder that contains two subfolders: One for the borrower rules and one for the loan rules.

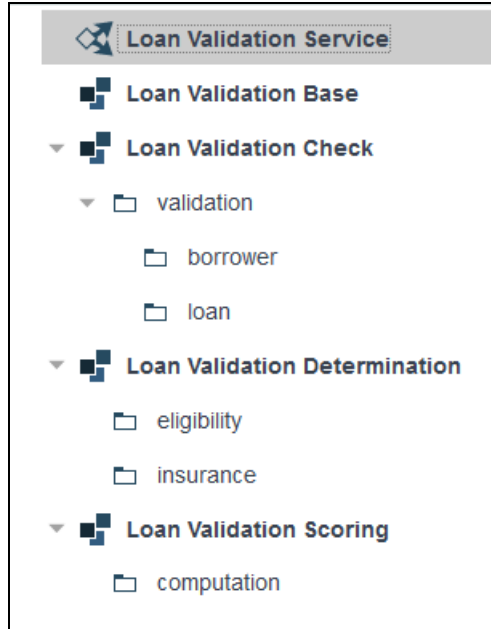


Figure 5-7 Folder structure of the Loan Validation Service

As an example of the structure by steps of the decision process, we first need to look at the ruleflow that orchestrates the overall execution of the rules in Figure 5-8 on page 70. Then we can go back to Figure 5-7 and see the folders corresponding to each of the rule tasks (validation, computation, eligibility, and insurance).

Tip: It is advised that each project has a unique folder hierarchy. This can easily be achieved by creating a uniquely named root folder.

Tip: When building ruleflows, it is not recommended to have rule tasks select individual rules or decision tables. The rule task should select the folder, so that if new rules are added to the folder, they are automatically included in the ruleflow.

5.4 Orchestrating ruleset execution

Rules apply decisions, but they do not have a defined sequence or succession. A ruleflow organizes rules into a sequence of decisions by assembling the rules into a group of rule tasks that uses a set execution pattern.

Each rule task is evaluated to produce a result, or decision. All of these results and decisions are combined to produce a single business decision, which is represented by a ruleflow. Ruleflows also specify conditional transitions between rule tasks.

The transitions determine how, when, and under what conditions to use each rule task (Figure 5-8).

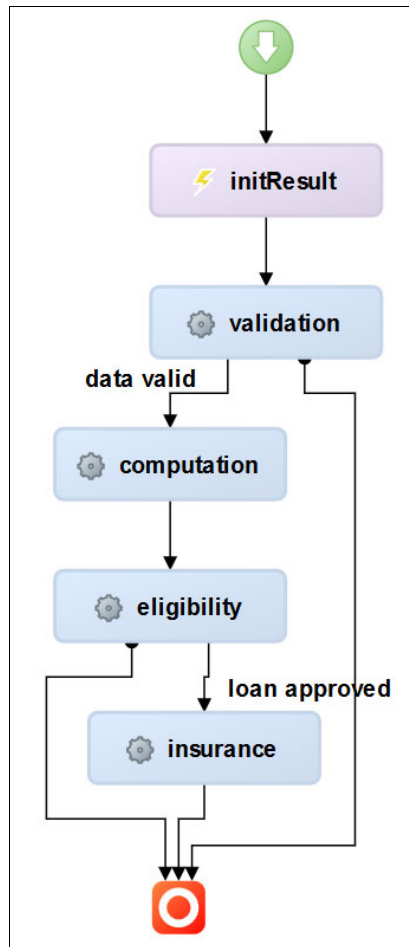


Figure 5-8 Ruleflow of the Loan Validation Service

5.5 Advanced topics

This section provides additional details about some of the more advanced topics.

5.5.1 Adding projects

Refactoring projects is straightforward when performed in stand-alone Rule Designer, and ideally the design of the project structure is completed in Rule Designer, before being synchronized with Decision Center.

After a decision service has been synchronized to Decision Center, refactoring projects and adding new projects to a decision service can be a bit complex. This process is described here as a reference for readers who might want to experiment with the steps described earlier in this chapter.

Steps to add a project

We have described steps in the evolution of the design of a decision service, we think it is helpful to provide the steps required to add a project in a decision service when it is already synchronized with Decision Center:

1. Import the branch/release of the decision service that you want to add the new project to into Rule Designer from Decision Center.
2. Create/import the new project that you want to add to the release/branch of the decision service.
3. Right-click the new project and connect it to Decision Center.
4. Publish this new project to Decision Center (this creates it in the main branch).
5. Add the dependency to this new project to the main project of the previously imported decision service.
6. Synchronize the decision service again, publishing the new dependencies (the `.ruleproject` file update).
7. Refresh (leave the Decision Artifacts page and return).

5.6 Conclusion

This chapter provided guidance on designing decision services so that your project can evolve as easily as possible over time. We learned that the following general practices help accomplish this goal:

- ▶ Separate the BOM project, Rules projects, and Decision Service project to provide the best flexibility over time.
- ▶ It is easier to manage permission at the project level, which means that separating rule projects by business domain is a good practice.
- ▶ Rule packages and folders are used to group rules that are logically related so that it is easier for an author to locate the rules they need to work with.
- ▶ Establishing the design of your project organization needs to be done early and in Rule Designer to avoid later refactoring.



Processes

IBM Operational Decision Manager (ODM) provides powerful capabilities to govern changes to business rules, but does not prescribe the precise orchestration of these capabilities. It is up to individual organizations to implement these processes. Processes describe the interaction between actors in the decision governance workflow. Actors could be human or automated (scripted). Processes also implement the integration of the decision governance workflow with IT Governance.

The goal of processes is to manage and control business policy change, and to ensure that all stakeholders are united in delivering IT and business releases. There are three process types:

- ▶ Processes that define business releases, including activities such as rule authoring, rule testing, rule deployment, rule retirement
- ▶ Processes that require technical changes, including activities such as Java execution object model (XOM) maintenance
- ▶ Processes that require a mix of business and technical changes

Processes for your organization can be affected by the following factors:

- ▶ Your users, roles, and responsibilities.
- ▶ Whether your processes are technical or business led.
- ▶ The “safety” of your rules. Can they be deployed directly by business users, or do they go through a full system test cycle before deployment?
- ▶ Service level agreements (SLA) for change. Are changes delivered the same working day, or are they delivered in monthly or quarterly cycles?

We explore some suggested processes to give you a starting point for defining your own initial IT release processes, business processes, and technical change release processes.

This chapter covers the following topics:

- ▶ Introducing the business release process
- ▶ Documenting processes
- ▶ Sample processes
- ▶ The IT Centric Process
- ▶ The business release
- ▶ The technical release
- ▶ Conclusion

6.1 Introducing the business release process

A *business release* contains one or more changes to the business policy. Business releases contain *change activities* and *validation activities*. A single business release contains one or more change activities and zero or more validation activities, as illustrated in Figure 6-1.

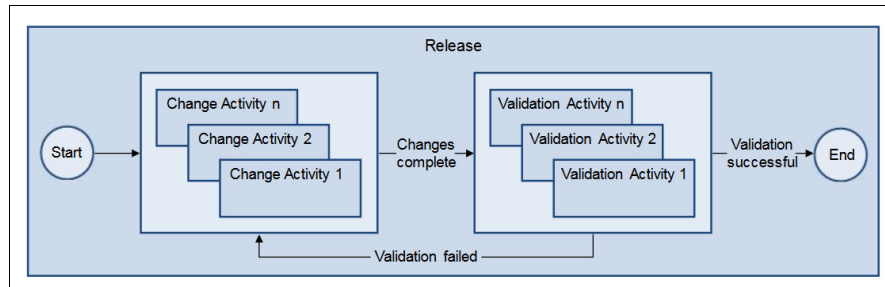


Figure 6-1 Components of the business release

A *change activity* is a modification to a single business policy approved by the change control board (CCB)¹.

A *validation activity* determines the test criteria for the change activities within the release. When all validation activities have passed, the release can be approved and deployed. A single release contains zero or more validation activities.

6.1.1 Change activities

A change activity is the smallest management unit of change. It is initiated by the Change Control Board in the following situations:

- ▶ Response to market conditions
- ▶ Response to new requirements and regulations
- ▶ Corrections to existing decisions

These situations initiate a change activity that is justified, prioritized, and tracked by the business to provide visibility and traceability. A leading practice is that a change activity should be as granular as possible, and that it corresponds to a single goal.

Each change activity can affect one or more business rules. The change activity has a status that provides a view of where this change activity is in its lifecycle. See “Change activity lifecycle” on page 87 for more details.

¹ In SCRUM terminology, it is the Business Owner who creates change activities after reviewing the backlog.

6.1.2 Validation activities

Validation activities ensure that the goals of the change activities are met and verified. If the validation fails, the change activities are rejected and the activity is revisited. A validation activity is not required if you choose to use the approval step within the change activity to perform validation.

The following validation activities exist:

- ▶ Peer review, including verifying the difference between current and previous versions of the rules from an IT and QA point of view.
- ▶ Rule review and validation from a business point of view, verifying that the rule meets the goals defined in the change activity.
- ▶ Simulation and comparison of the results to previous releases to ensure that the results are expected.
- ▶ Non-regression tests.
- ▶ User acceptance tests.
- ▶ Integration testing with external systems that interact with the decision service.

Validation activities are addressed in more detail in section 6.1.2, “Validation activities” on page 75.

6.2 Documenting processes

The documentation of processes helps formalize the communication and responsibilities between the actors involved in the processes. It provides traceability for auditing and reporting. It ensures that information flows faster to more people.

The documentation can also be used as a communication and training tool for new team members. It reduces the knowledge loss when other team members leave. Documented processes provide a baseline that can be used for process improvement for your organization.

6.3 Sample processes

There are many processes to consider in the governance of decisions, and this chapter provides examples that can be used as a starting point for organizations to build their own internal processes. The processes shown in the following sections can be used as inspiration for the change processes your organization can put in place.

To keep it as simple as possible, assume that a release contains only one activity. Then we can enlarge and consider a process containing multiple activities.

Consider some possible processes for each type of release discussed earlier:

- ▶ Process to create the initial release
- ▶ Process for subsequent IT releases
- ▶ Process applying to the business release
- ▶ Process applying to the technical release

6.4 The IT Centric Process

In the IT Centric process, we distinguish between the *initial release*, and subsequent IT releases.

Notes about the initial release: The first publishing of a decision service to Decision Center creates an initial, closed release in Decision Center.

In a Business Centric process, subsequent changes to the decision service in Decision Center take place in releases that are created from this initial release. The initial release is not typically deployed to production because it has not been tested by the business.

In an IT Centric process, the initial release may be deployed to production because it was typically tested before being synchronized with Decision Center.

6.4.1 The initial IT release

The initial release is IT Centric and is the starting point of the decision service lifecycle. It contains initial business rules, decision service, and supporting application infrastructure.

The IT department consults with the business to ensure that the rules are understood and maintainable.

This release might not necessarily be the first live deployment, because the business might need to make further changes in Decision Center and produce a business release before going live.

The initial release is the starting point of your decision service lifecycle, and is required whether you choose to implement decision governance or IT governance.

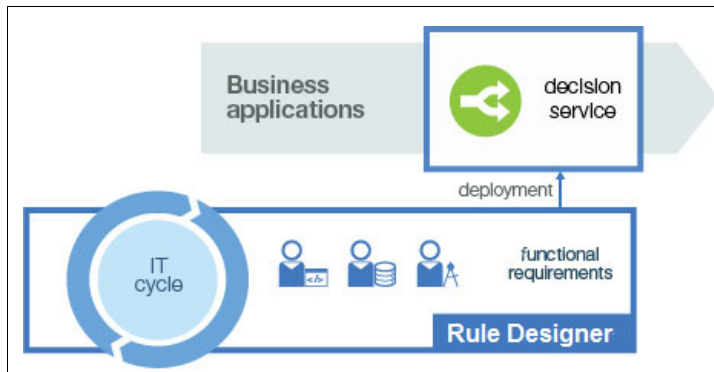


Figure 6-2 The initial release

The main contributor of the initial release is the Rule Developer who works in Rule Designer, as illustrated in Figure 6-2. They develop the technical rule artifacts, the rules, and the decision service. The first version of the decision service can be deployed from Rule Designer to Rule Execution Server. In this release, business users access the rules in Decision Center, but in view mode only. They are not granted the permission to make any changes.

This phase is the right time to practice and tailor the decision governance process, and to train the business to ensure that they are ready for making changes.

We can formalize the steps for such an IT Centric cycle:

1. The business SMEs define the rules matching the business policy, and pass the requirements to the Rule Developers.
2. The Rule Developer develops rules and the decision service in Rule Designer.
3. The Rule Developer publishes the decision service to Decision Center for business users for review.
4. Business users access the rules in Decision Center, but in read only mode.
5. When business users identify a need to change the rules, they notify the Rule Developer by using requirement documents. The Rule Developer implements the changes in Rule Designer, and then publishes the updates to Decision Center, where the business users have an up-to-date copy of the rules. Versions are maintained in Decision Center this way, but change is driven from Rule Designer.
6. The decision service is deployed for the first time from Rule Designer to Rule Execution Server.

Note: The previous sequence of steps can slightly differ if we consider, in the decision governance perspective, as described in Chapter 7, “Decision governance framework” on page 85, that the business users validate the initial IT release by performing validation activities in a new release created from the initial IT release. Refer to section “7.2.2, “The first IT release” on page 90 for details.

The previous sequence of steps is illustrated in Figure 6-3.

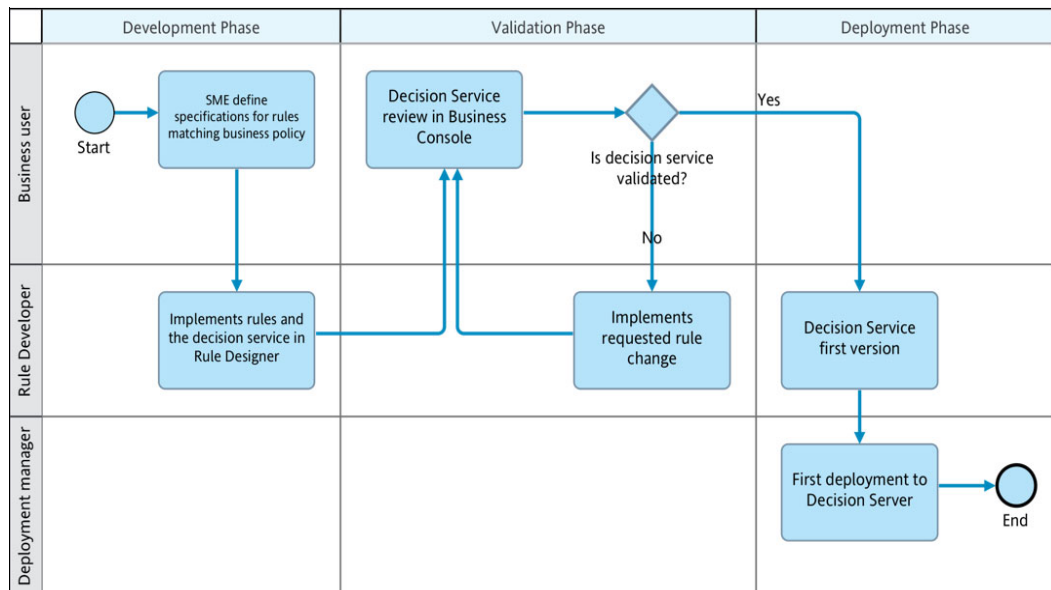


Figure 6-3 Steps of initial release Creation

In the initial IT release, because all rule authoring and management is performed from Rule Designer, this IT Centric cycle is appropriate for Rule Developers familiar with programming languages and development environments. There is only IT governance in the initial release, so there is no role-based permission management for business users.

6.4.2 Staying in IT: IT release for subsequent releases

If your organization chooses to use IT governance for all subsequent changes, Rule Designer remains the tool for all releases. The rules source of “truth” is the source code control

repository. If the business users identify a need for changes, they provide their feedback to IT by using requirement documents. IT processes the changes in Rule Designer in subsequent releases. Each new release is built upon the previous completed release.

The different steps of building a technical release in an IT Centric cycle are shown in Figure 6-4.

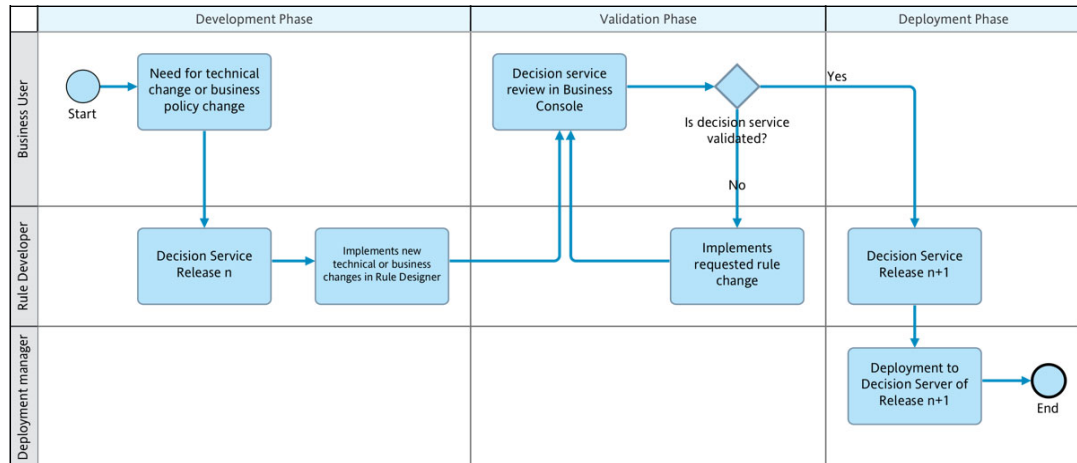


Figure 6-4 Subsequent technical release process

6.5 The business release

The *business release* is where the business takes over from IT after the initial release. Each business release is enforced by the decision governance features of ODM, and follows a strict workflow to ensure that changes to business policies are constrained, tested, and correct. Business releases can be deployed frequently to reflect up-to-the-minute business requirements.

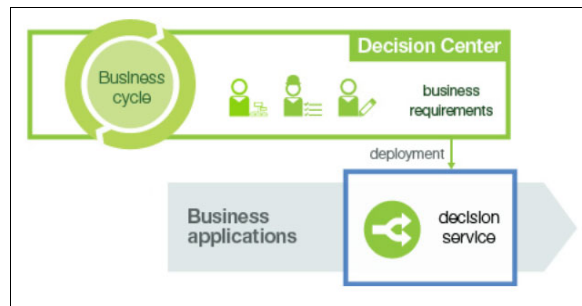


Figure 6-5 The business release

The main actors of the business release are the business owner and the team. The team can consist of business analysts and rule authors, or just rule authors.

The business owner and the team reviews the rules from the Business Console, and if they identify the need for change, they change rules in the Business Console, given their access rights. These depend on the permissions enforced in Decision Center. Decision Center is the source of “truth” for the rules, and the business drives deployment to Rule Execution Server from the Business Console, as illustrated in Figure 6-5.

We can formalize the following steps for such a business release:

1. The initial release was created, as shown in 6.4.1, “The initial IT release” on page 76, and a new release was created from this initial release.
2. The business user updates and creates rules in the Business Console.
3. The deployment manager deploys the decision service to a test Rule Execution Server where business users validate the rules.
4. After validation is complete, the deployment manager deploys the decision service to Rule Execution Server from Decision Center or, more typically, generates a decision service archive from the Business Console and places it in a shared location where the deployment team then deploys it to the Rule Execution Server.

In this type of release, because all rule changes are driven from the Business Console, there is no possibility to modify the XOM, so rule changes cannot concern any model update or return type changes. Only rule logic changes can be handled, based on an existing BOM model.

6.5.1 Simple business rule change

The process in Figure 6-6 shows the minimum set of activities required to implement a simple business rule change.

For example, this process is used to make a simple change to a rule or a decision table. For example, if a production issue is found and must be fixed quickly. A new release, called “Bug Fix-xx Release” will be created from the current release, and the fix will be implemented in this release. The creation of the release is not represented in the diagram. Only the development and testing activities are represented.

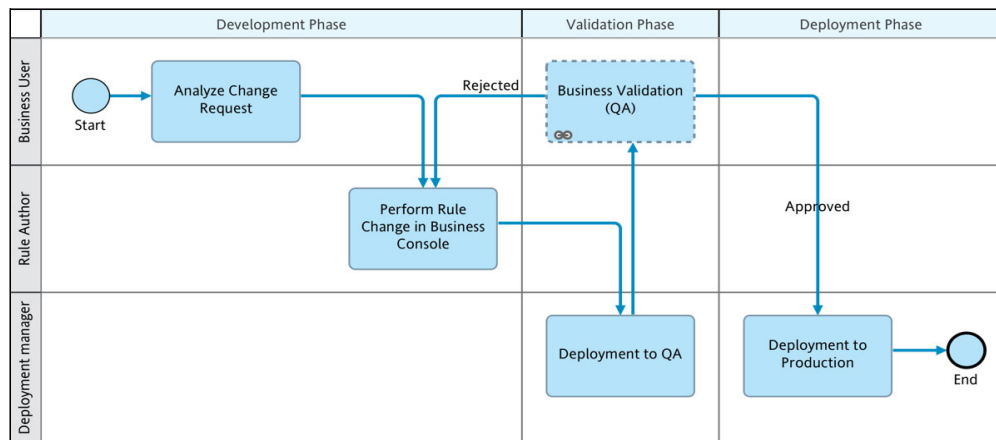


Figure 6-6 Business simple rule change process

Here are some notes about this process:

- The change activity is assumed to include the rule author’s validation to prove the change works.
- The business validation activities can only start after all of the change activities are approved.
- The deployment of the test fix occurs immediately after the validation is approved.
- The change is considered simple enough to not require a second validation step before deployment to production.

6.5.2 Complex business rule change process

This process describes activities to be undertaken if the release requires complex changes to business rules, but does not require technical support from IT.

The complex business rule change process includes an extra validation step performed in a user acceptance testing environment, as shown in Figure 6-7. Some organizations might have a different second validation step, but *multiple validation* is a common practice.

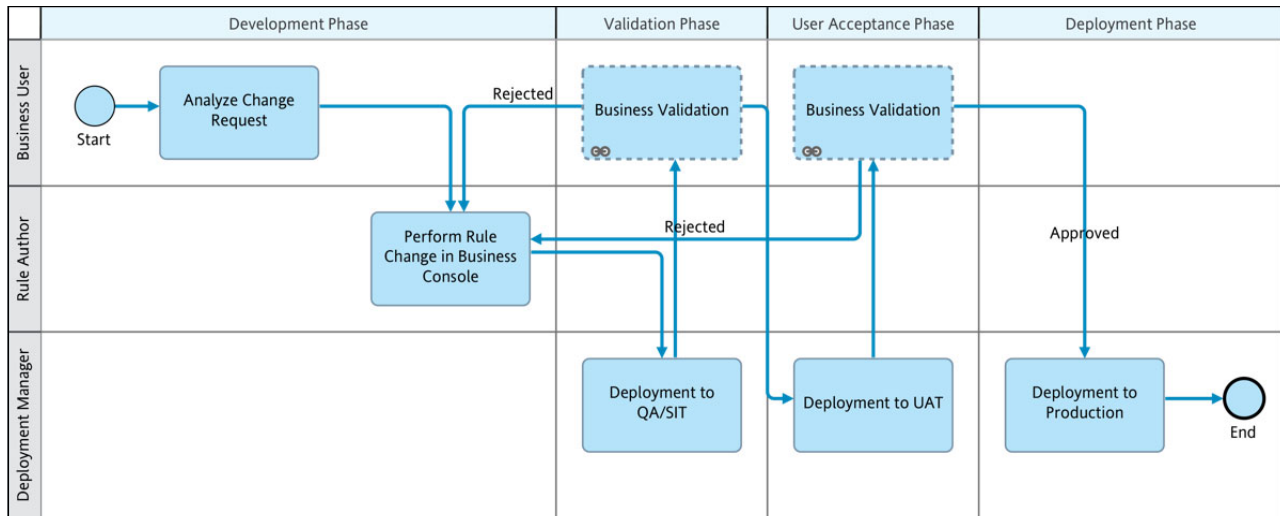


Figure 6-7 Business complex rule change process

6.6 The technical release

At some point, technical changes need to be applied to the decision service, such as enhancements to the Business Object Model or to upgrade ODM. In its simplest form, this cycle requires the business to stop work and allow IT to synchronize their changes to Decision Center. If regular business releases are required, this is not feasible. A more realistic approach is to allow both business and IT to work in parallel, as shown in Figure 6-8, and use the advanced synchronization and merging capabilities of Decision Center to handle conflicts between IT changes and business changes.

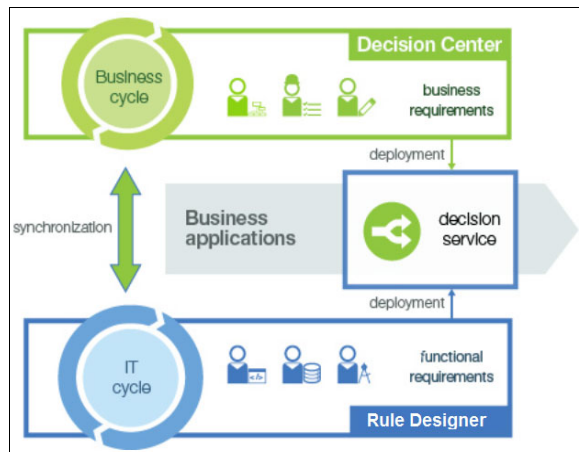


Figure 6-8 The technical release

The following actors are the main actors in this type of release:

- ▶ The Rule Developer, who performs and implements technical changes in Rule Designer
- ▶ The business users (rule authors, business policy analysts), who continue to apply business policy changes in Decision Center.

At some point, when the Rule Developer has sufficiently debugged and tested the decision service in Rule Designer, he synchronizes the decision service with Decision Center and use the merging features of Operational Decision Manager to reconcile his changes with the changes that the business team has made. Decision Center allows fine-grained reconciliation between the IT and the business versions. Specifically, it offers capabilities to update, override and update, publish, or override and publish each project element that has changed.

Figure 6-8 on page 80 shows the process that you can formalize with the following steps:

1. Before performing any change in Rule Designer, the Rule Developer synchronizes (updates) the latest decision service projects from Decision Center to his development environment.
2. The Rule Developer implements technical rules changes in Rule Designer, for instance performing XOM changes or updating the rule flow.
3. The Rule Developer publishes the decision service to Decision Center in a new release called *IT release*, for business users' review.
4. In the meantime, business users might have been creating additional releases in Decision Center.
5. Business users review the latest technical updates and, if they are happy with them, merge their latest business release to the IT release. If the changes are complex, they might require assistance from IT.
6. After the Business changes have been merged with the IT changes, the IT release goes through the regular testing cycle before being deployed to the production environment.

The previous steps are represented in Figure 6-9 on page 82 and Figure 6-10 on page 83. As the resulting diagram is wide, it was split into two diagrams for clarity of reading. Figure 6-9 on page 82 shows the development phase, while Figure 6-10 on page 83 shows the following validation and deployment phases. Links were used in the diagrams to show the next steps.

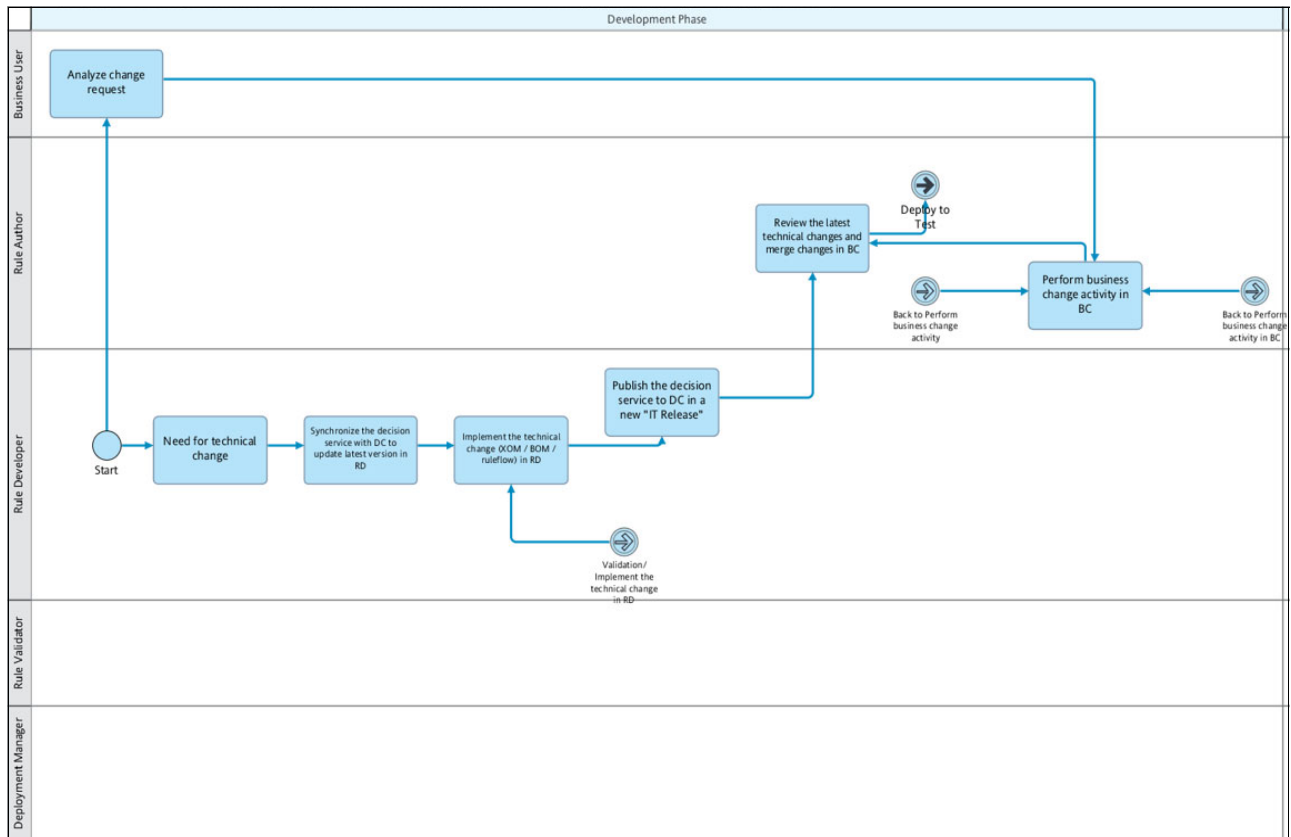


Figure 6-9 Technical release Process, Development Phase

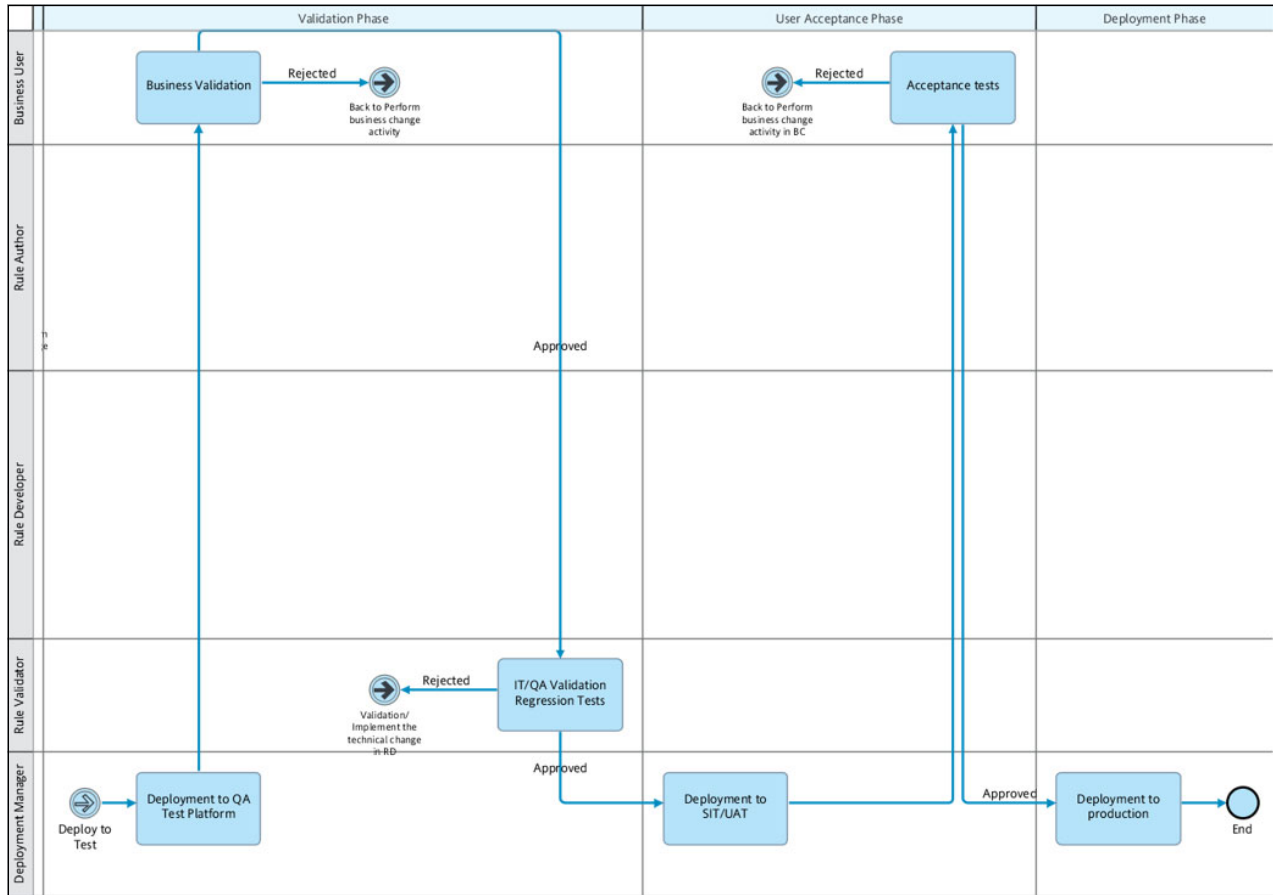


Figure 6-10 Technical release Process, Validation and Deployment Phase

6.7 Conclusion

We have provided four types of processes that can occur in changing business rules, namely the initial release, the IT Centric release, the business release, and the technical release. The processes shown in this chapter can be adapted by readers to make them appropriate for their organization. Some organizations might decide to add more steps and create additional processes to handle the reality of their environment.



Decision governance framework

IBM Operational Decision Manager (ODM) provides the decision governance framework to manage and organize the decision services lifecycle. The decision governance concepts were introduced in Chapter 6, “Processes” on page 73.

This chapter is about the decision governance framework implemented with the Business Console. The framework enforces decision governance and follows a strict decision governance process to ensure that changes to business policies are constrained, safe, and correct.

The decision governance framework encompasses the following concepts:

- ▶ *Releases* are categorized as *IT release* when Rule Developers using Rule Designer are involved, and *Business release* only when rule authors working from the Business Console are involved. Completed releases are deployed to the Rule Execution Server.
- ▶ *Change activities* are where the Rule Author or Rule Developer change rule artifacts to implement the decision service release.
- ▶ *Validation activities* track the execution of tests and simulations validating the change activities.

This chapter provides the following information:

- ▶ Explains how the Business Console organizes and coordinates the project team.
- ▶ Explains decision service project management using decision governance framework releases and activities.
- ▶ Details a typical sequence of releases, as shown in Figure 7-1 on page 86.

This chapter covers the following topics:

- ▶ Decision governance framework lifecycle
- ▶ A decision governance framework workflow pattern
- ▶ Loan Validation Service illustration
- ▶ Conclusion

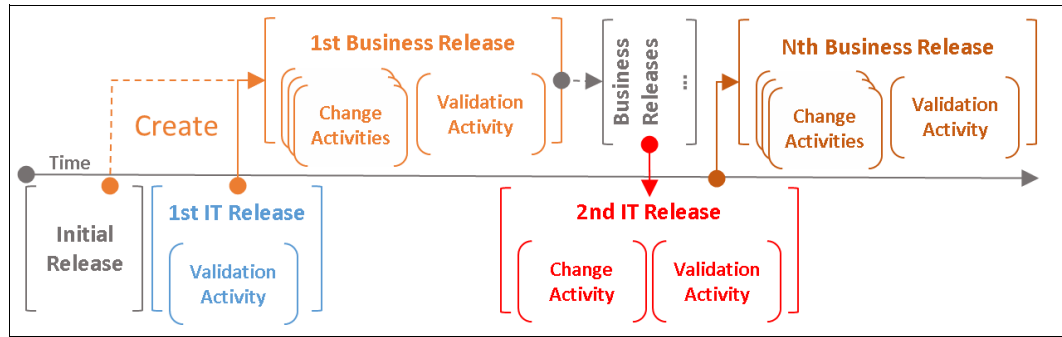


Figure 7-1 Typical sequence of releases

7.1 Decision governance framework lifecycle

The decision governance framework lifecycle is the sequence of release, change activities, and validation activities leading to a production business release.

For each release created, at least one change activity or validation activity must be created.

The individual lifecycle of each release and its change activities and validation activities depends on its status and the status of the other Decision Center Governance elements.

In this section, we detail the lifecycle of releases, change activities, and validation activities.

7.1.1 Release lifecycle

A release holds every decision service artifact described in Chapter 5, “Designing decision services” on page 61. The release status is the expression of work progression towards its goal state of complete and ready for deployment, Figure 7-2 on page 87.

- ▶ *In Progress* is the initial status after creation and implementation work is necessary before being ready to be deployed to production.
- ▶ *Ready for Approval* status means that all changes are completed and are waiting for release approval before switching the status to *Complete*.
- ▶ *Complete* is the final status of the release after every change is completed and tested. This state allows the decision service to be deployed to the production Rule Execution Server.
- ▶ When not approved, the status becomes *Rejected* before reopening the release to pursue the release implementation.
- ▶ A release can be *Canceled*.

The following users are involved in release management:

- ▶ The *owner* manually changes the release status from *In Progress* to *Ready for Approval*, *Canceled*, and possibly back to *In-Progress* after being reopened.
- ▶ The *approvers*, of which there can be several, review and test the changes when the release is in status *Ready for Approval*. If any of the approvers rejects the release changes, the release is *Rejected*. Otherwise, it ends up in state *Complete* and is ready for deployment.

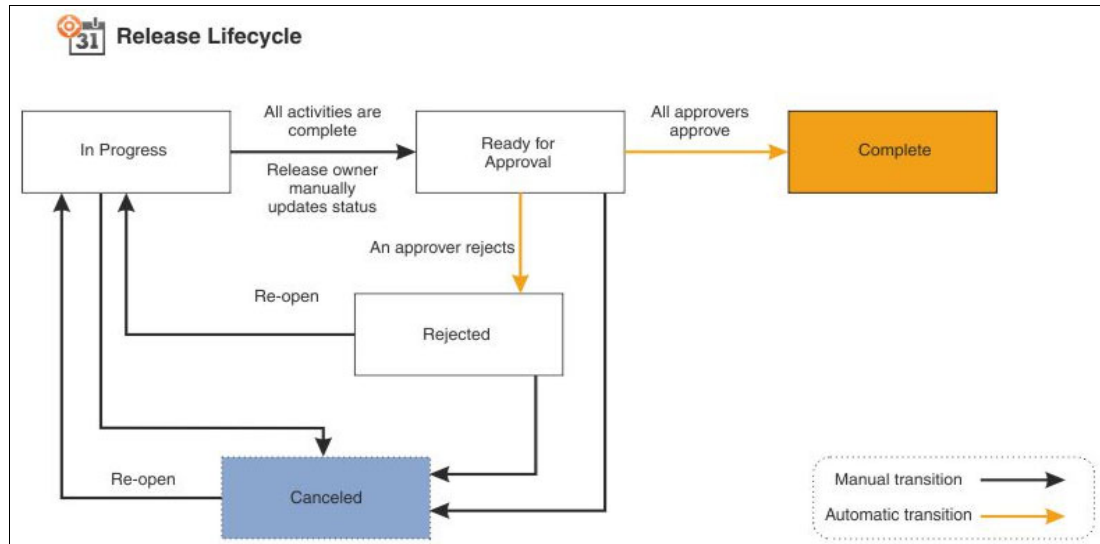


Figure 7-2 Release lifecycle

The decision governance framework allows several releases to be concurrently in progress.

Apart from the initial release, releases are always based on a parent release. They are created with a name, a goal, a due date, an owner (the creator) and one or several approvers.

Tip: A release is a decision service container and cannot be used to edit its content directly. Change activities attached to the release are designed to organize the release implementation.

7.1.2 Change activity lifecycle

Change activities are required to reach a given release completion.

As with the release, the change activity lifecycle is the expression of its status changes, Figure 7-3 on page 88:

- ▶ *In Progress* when created, rejected by approver, or reopened after being canceled.
- ▶ *Ready for Approval* from *In Progress* when every author declares their work finished.
- ▶ *Complete* when change activity approvers verified and approved the modifications.
- ▶ *Canceled* when the planned change activity is abandoned.

The following users are involved in a change activity:

- ▶ The owner of the change activity.
- ▶ The authors responsible for creating and updating the rules.
- ▶ Only authors assigned to a change activity that is *In Progress* can create or change project elements.
- ▶ The approvers verify the changes after the authors' work is complete.

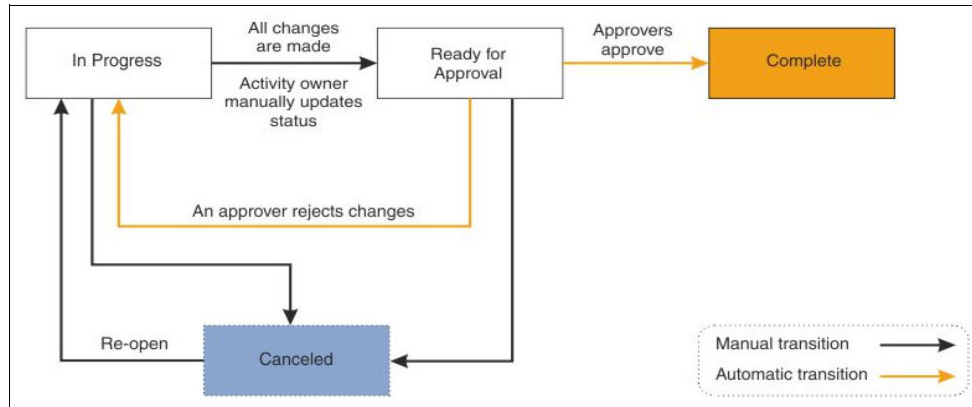


Figure 7-3 Change activity lifecycle

There can be several concurrent on-going change activities. When approving a change activity after a prior change activity was already completed, a merge may be necessary to resolve conflicts if the same artifacts have been updated.

Tip: It is a good practice to avoid modifying the same artifacts in two parallel change activities. In fact, when an artifact has been modified in one change activity, it is locked and authors are forced to coordinate the changes to avoid conflicts.

When all of the release change activities are complete, Rule Testers can start validating the changes within validation activities.

7.1.3 Validation activity lifecycle

Validation activities are optional to reach a given release completion.

As with the change activity, the validation activity lifecycle is the expression of its status changes, as shown in Figure 7-4 on page 89:

- ▶ *In Progress* when created, rejected by an approver, or reopened after being canceled.
- ▶ *Ready for Approval* from *In Progress* when every tester declares their work finished.
- ▶ *Complete* when the validation activity approvers verified and approved the tests and simulations.
- ▶ *Canceled* when the planned validation activity is abandoned.

The following users are involved in a validation activity:

- ▶ The owner of the validation activity.
- ▶ The testers responsible for creating and running the tests and simulations.
- ▶ Only testers assigned to a validation activity that is *In Progress* can create and run tests and simulations.
- ▶ The approvers who verify the changes after the testers' work is complete.

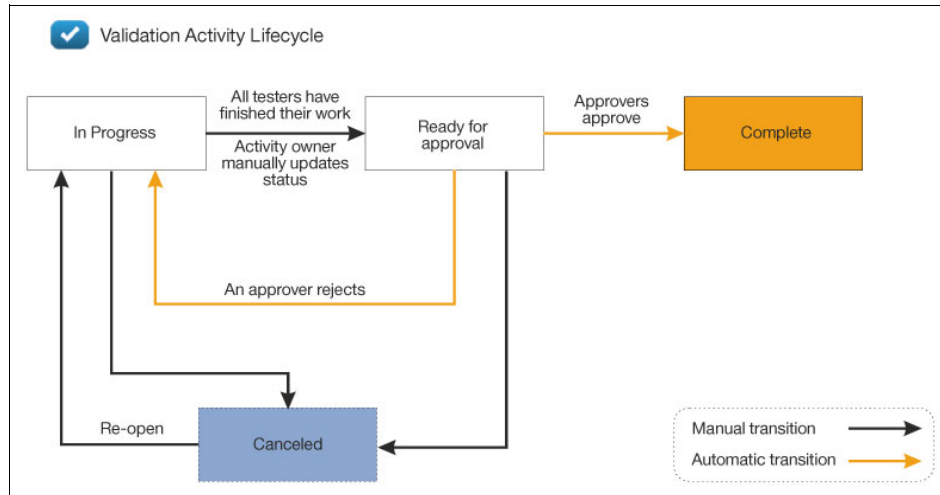


Figure 7-4 Validation activity lifecycle

7.2 A decision governance framework workflow pattern

This section describes a sample decision governance framework workflow pattern.

7.2.1 The initial release

The initial release is created by synchronizing a new decision service from Rule Designer to Decision Center, as shown in Figure 7-5.

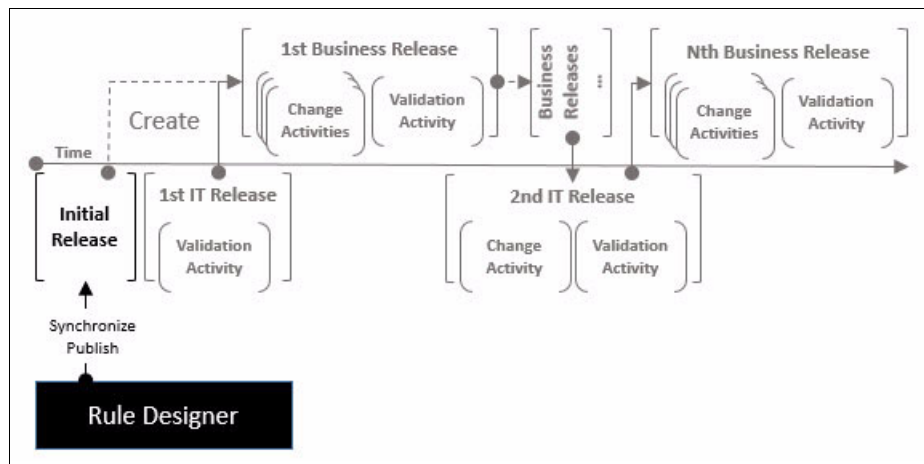


Figure 7-5 The creation of the initial release

Because Rule Designer must be used to create the initial release, it involves IT and Rule Developers.

This initial release is critical because it creates the root decision service design on which all subsequent releases are based. The critical components of the decision service design are the business object model (BOM), the project hierarchy, and the decision service operations signature, as described in Chapter 5, “Designing decision services” on page 61.

Business users and ODM consultants must be involved in the design of this initial phase to make sure that the business requirements are properly met, and that the decision service design is flexible and robust.

In the case of a simple decision service, the initial release might be ready for production deployment straight from Rule Designer after testing by the technical team, but this is not recommended. A better approach would be to publish the decision service to Decision Center and deploy it to production as “The first IT release”.

Tip: As of version 8.9, a decision service using the Decision Engine cannot be tested using Decision Validation Service (DVS) Microsoft Excel sheets that are run within Rule Designer. If you plan to run DVS Excel tests, a new release started from the initial release must be planned to configure and run these tests from the Business Console.

Tip: Rule Developers using Rule Designer are required to create the decision service hierarchy, BOM, and vocabulary. All other artifacts can be created within the Business Console.

7.2.2 The first IT release

The first IT release is an optional and conventional release category useful to formalize the IT handover of a decision service to the business users. It is a child release of the ODM decision governance framework initial release, as shown in Figure 7-6.

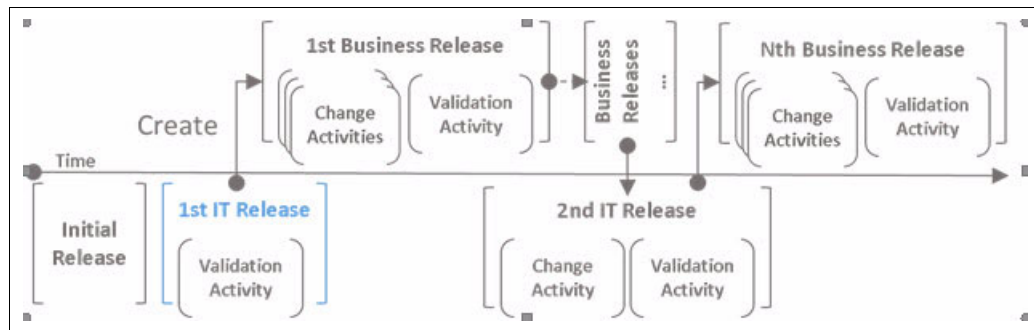


Figure 7-6 The initial IT release

Because it is impossible to add a validation activity to the completed initial release, this first IT release is necessary when business users are required to validate the first version of the decision service delivered by IT using Tests and Simulations within a validation activity. See Figure 7-7 on page 91.

The first IT release validation activities are also necessary when the decision service execution mode is Decision Engine, because tests and simulations based on scenarios using DVS Excel sheets can only be executed on Decision Runner from the Business Console.

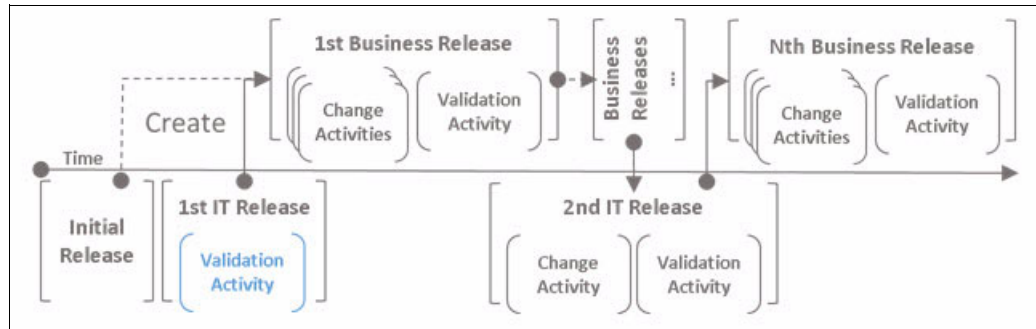


Figure 7-7 Validation activity in the first IT release

When the first IT release validation activity is approved, the release is complete and it can be forwarded to the business user involved in rule authoring. This is done creating a business release from the first IT release.

7.2.3 Validation activities

When all the release change activities are complete, rule testers can start validating the changes within validation activities (Figure 7-8).

Validation activities are optional, and a release may have several in the case of complex decision services where validation tests must be split between different teams responsible for different functional areas.

Validation activities might consist of simply verifying that the business rules conform to the requirements. In that case, the *tester* simply certifies that this verification was performed.

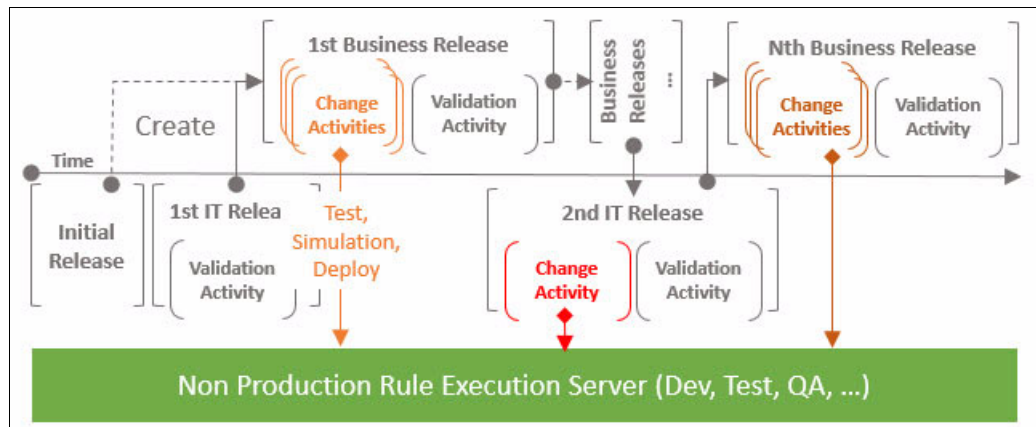


Figure 7-8 Validation activities deployed to non-production Rule Execution Servers

Validation activities are otherwise designed to create and run tests and simulations based on scenarios defined in Excel sheets:

- ▶ Testing is performed by running a set of rules on scenarios, and then comparing the actual results with the expected results.
- ▶ Simulations are performed by running rules on large sets of real or fictitious operational data, and using the results to assess and refine the behavior of the rules. Key performance indicators (KPIs) are applied in analyzing the rules.

Tests and simulations can also be created as part of change activities, and simply be run to verify that the changes conform to the expected results.

Tests and simulations can only be executed on Rule Execution Servers categorized as non-production.

When completed, every test and simulation report remains attached to the validation activities. These artifacts are the evidence supporting whether the validation of the release was successful.

When every validation activity is approved, the release can be completed.

7.2.4 Business releases

A *business release* is the release category tailored to the decision governance framework. A business release goal is to implement business rules until a complete status is reached and the decision service RuleApp and ruleset can be deployed in production (Figure 7-9).

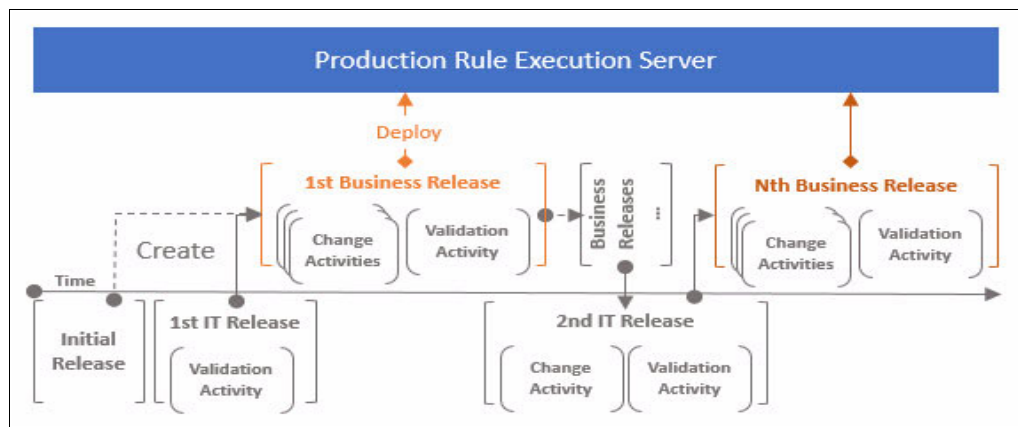


Figure 7-9 Business release deployment to Production Rule Execution Server

A business release, as for any release, consists of at least one change activity.

Optional validation activities can be created when changes are complete in the change activities.

The business release is a conventional term implying that changes are only managed by Business Rule Authors using Business Console.

7.2.5 Change activities

The goal of a change activity is to manage decision service artifacts and tests, simulate and deploy the resulting RuleApp on a non-production Rule Execution Server (Figure 7-10).

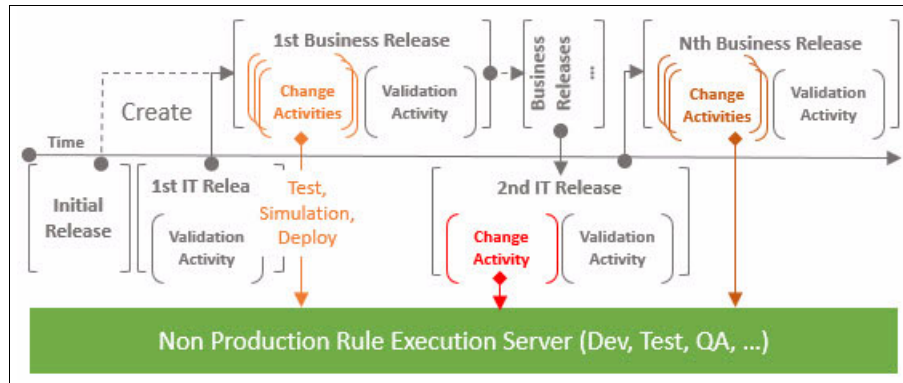


Figure 7-10 Testing change activities on non-production Rule Execution Server

In case of business release change activity, the project security configuration should only enable Business Rule Authors to manage Action Rules, Decision Tables, Tests, and Simulations as described in Chapter 4, “Securing the Decision Center” on page 31.

If Rule Developers are also authors of the change activity, project security may allow them to also manage all decision service artifacts.

When change activities are completed, validation activities can be created and performed. This was described in “Validation activities” on page 91.

If the scope of the planned changes is wide, it is possible to create several change activities where the workload will be distributed across different teams. Validation activities must be created, which forces tests and simulations to be performed when all changes are completed. This process ensures that the tests are performed on the resulting merged changes.

Tip: Make sure that no change is common to two *In Progress* change activities to prevent conflict. A rule that is being edited in a change activity is locked until the activity is completed and merged into the release.

7.2.6 The IT release

The IT release is an optional and conventional category of release useful to formalize the IT handover of a technical change to an existing decision service to the business users.

An IT release is necessary when the rule projects of the decision service are reorganized or the Business Object Model is modified. These changes can only be made through Rule Designer, and it is likely that the time required to perform these changes will be longer than for a business release. IT releases are therefore parallel to the In-Progress business release (see Figure 7-18 on page 95).

When complete, the IT release will be the base release of the next new release.

Because IT releases are essentially managed from Rule Designer in the IT domain, the process is as follows:

1. Create the IT release based on the latest Business release, as shown in Figure 7-11.

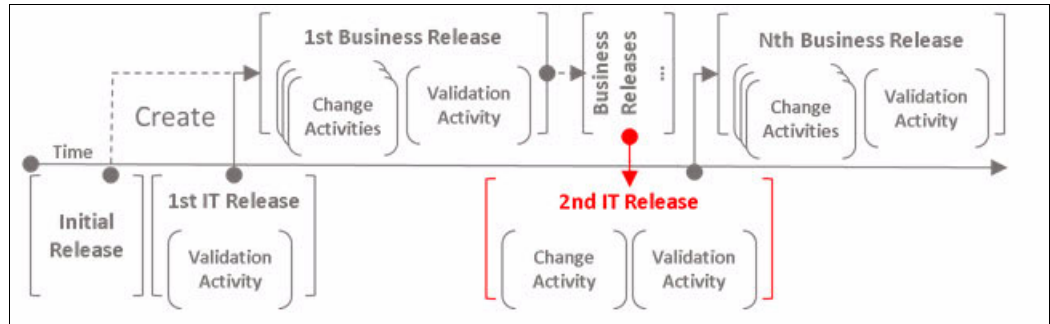


Figure 7-11 IT release based on the latest Business release

2. Initialize a Rule Designer workspace from the IT release, as shown in Figure 7-12.

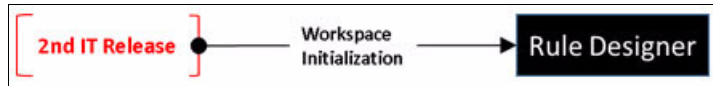


Figure 7-12 Rule Designer workspace

3. Create a new rule project in Rule Designer (*no decision service artifact*).
4. Create or synchronize the new rule project to Decision Center (creation of a main branch), as shown in Figure 7-13.

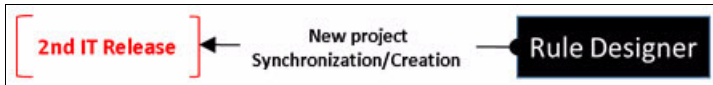


Figure 7-13 New rule project

5. Publish the .ruleproject decision service to the IT release (Figure 7-14).

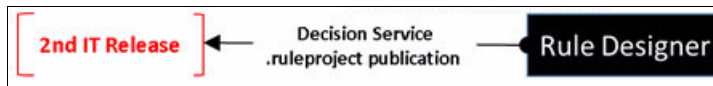


Figure 7-14 Publication of the decision service

6. Modify rule project dependencies.
7. Publish the modified rule project dependencies to the IT release, as shown in Figure 7-15.

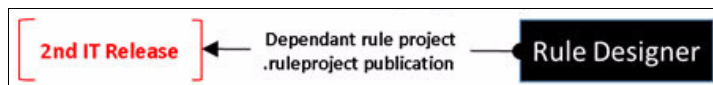


Figure 7-15 Modified rule project dependencies

8. Update the Rule Designer workspace with the most recent Business release, as shown in Figure 7-16.

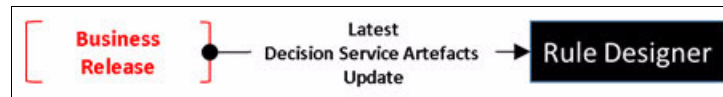


Figure 7-16 Update Rule Designer workspace

9. Create an IT release change activity.
10. Publish the decision service artifacts from Rule Designer to the IT release change activity, as shown in Figure 7-17.

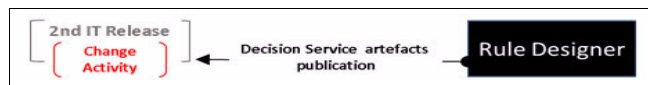


Figure 7-17 Publication of decision service artifacts

When you reach this last step, follow the base decision governance framework process to complete the IT release change activity and the IT release. When complete, the next business release is based on the most recent IT release, as shown in Figure 7-18.

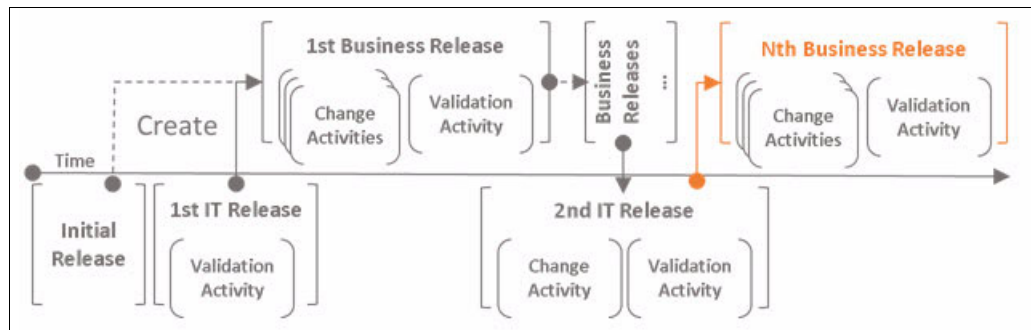


Figure 7-18 Creation of a Business release based on an IT release

Project security configuration should enable change activity authors to be Rule Developers who are allowed to change any artifact within the decision service (as opposed to business rule authors who should only be allowed to manage rules, tests, and simulations).

7.3 Loan Validation Service illustration

To conclude this chapter, we illustrate the decision governance framework workflow to implement the Loan Validation Service sample using the Business Console.

The Business Console is a web-based interface and has the following URL is by default:

`http://<server_name>:<port>/decisioncenter`

This section illustrates the following phases:

1. Creating the Loan Validation Service initial release
2. Creating the Loan Validation initial IT release
3. Creating the first business release, and completing change and validation activities

7.3.1 Creating the initial release

The following steps are involved in creating the initial release:

1. Ivan, the ODM developer, sets up the first version of the Loan Validation Service project from Rule Designer, as described in Chapter 5, “Designing decision services” on page 61. It is a simplified decision service project with rules, BOM, and Java XOM to demonstrate the releases and activities lifecycle, as shown in Figure 7-19.

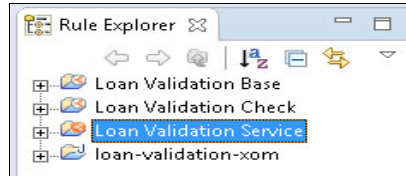


Figure 7-19 Rule Designer Loan Validation Service project

2. He can then connect to the Decision Center by right-clicking the **Loan Validation Service** and selecting **Decision Center** → **Connect**, as shown in Figure 7-20.

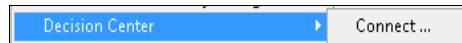


Figure 7-20 Rule designer Loan Validation Service connect to Decision Center

3. Adam, the ODM administrator, provides the necessary connection data (Figure 7-21).

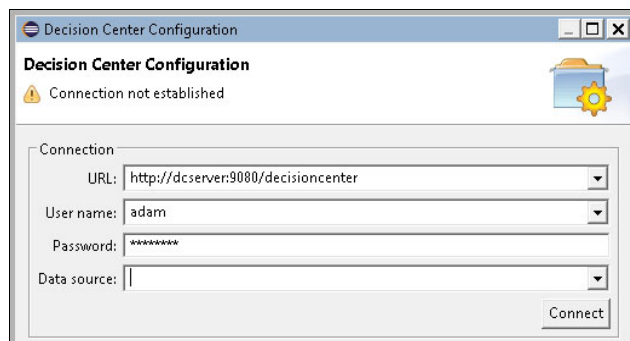


Figure 7-21 Rule Designer Loan Validation Service synchronization

4. It is critical here to select **Use Decision Governance Framework** to make sure to manage this decision service following the ODM decision governance workflow within the Decision Center, as shown in Figure 7-22.

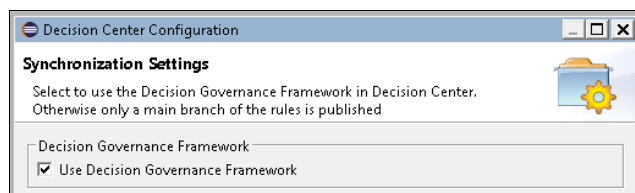


Figure 7-22 Rule Designer Loan Validation Service synchronization

5. The necessary dependent projects are automatically added to Decision Center, as shown in Figure 7-23.

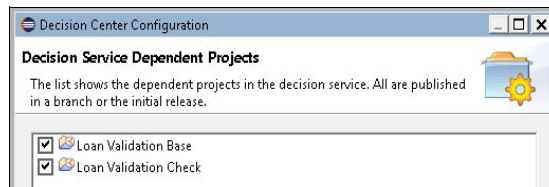


Figure 7-23 Rule Designer Loan Validation Service synchronization

6. Because it is a new decision service, no additional conflict resolution between Rule Designer and Decision Center is necessary when synchronization to Decision Center is complete. The message Synchronizing: No changes found displays (Figure 7-24).

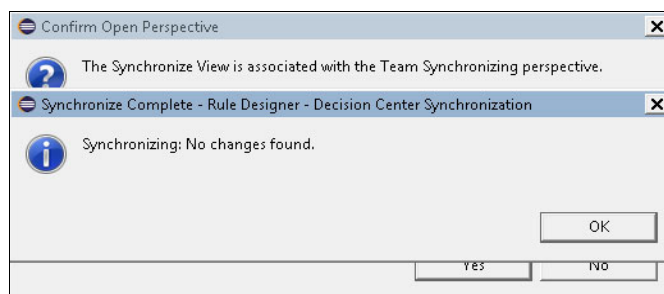


Figure 7-24 Rule Designer Loan Validation Service synchronization

7. Adam can now connect to the Business Console, Figure 7-25.

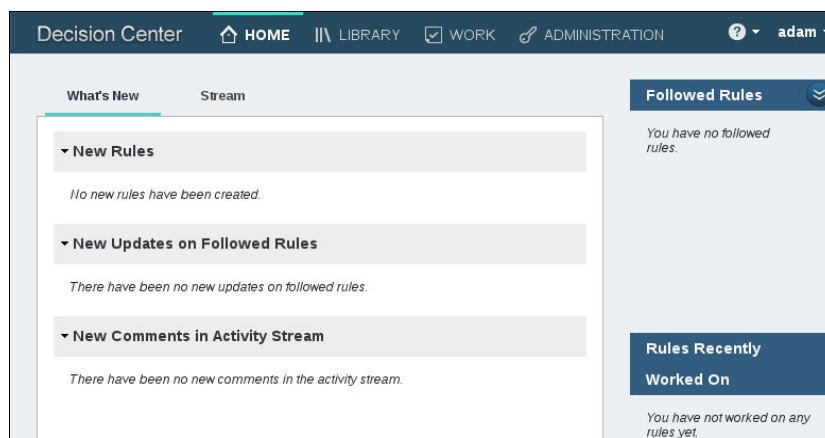


Figure 7-25 Decision Center Home tab

8. Verify in the Library tab that the Loan Validation Service has been created, as shown in Figure 7-26.

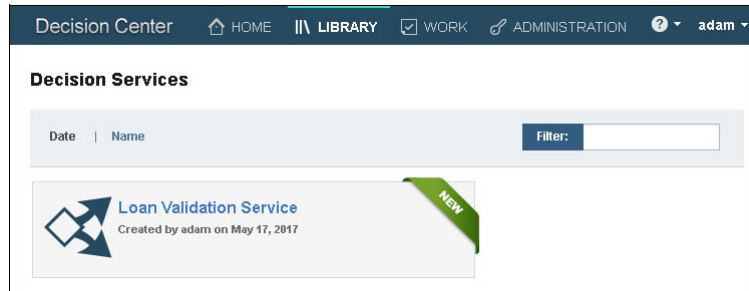


Figure 7-26 Decision Center Library tab

9. Also verify the initial release, when selecting the Loan Validation Service, as shown in Figure 7-27.

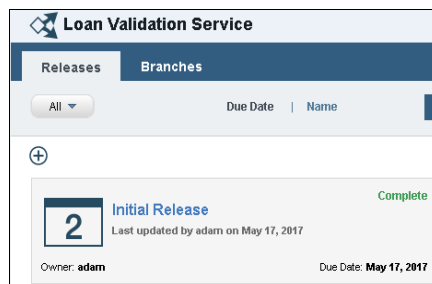


Figure 7-27 Loan Validation Service Initial Release

7.3.2 Creating the first IT release

Before handing over the decision service to the business user to complete rule development, Adam, the ODM administrator, creates a first IT release, as described in “The first IT release” on page 90. He also creates a validation activity to demonstrate that Loan Validation Service typical scenarios are working as designed:

1. From the Decision Center Library tab, Adam clicks the Plus sign (+) or the **arrow** next to Initial Release (Figure 7-28) and then sets Paul, the release manager, as an approver, as shown in Figure 7-29 on page 99 and Figure 7-30 on page 99.

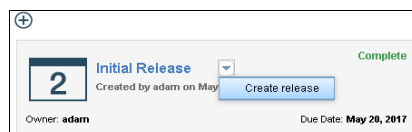


Figure 7-28 Step one

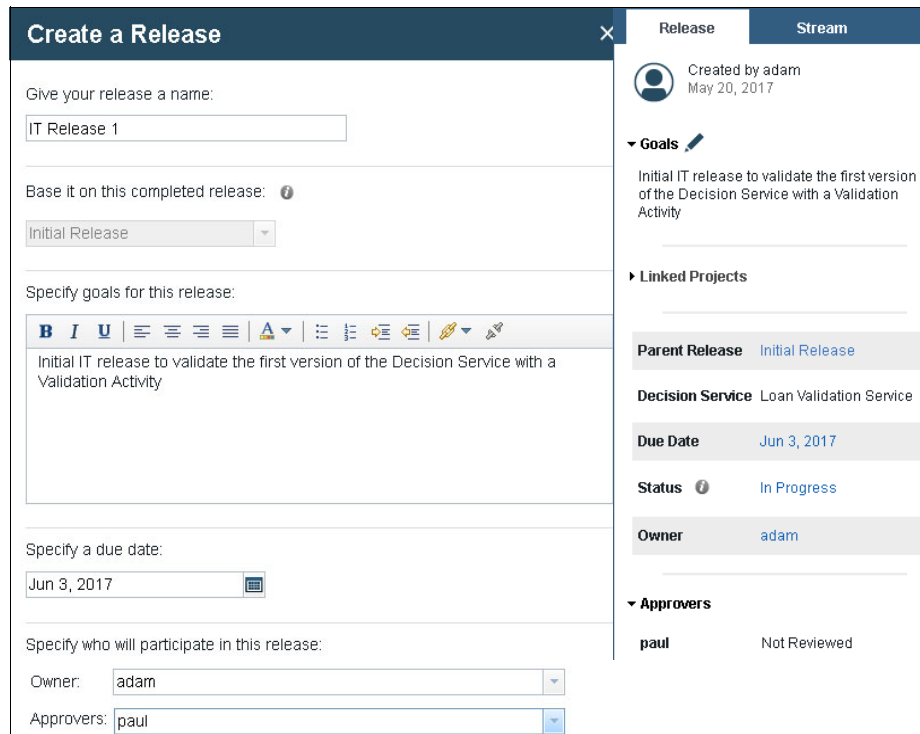


Figure 7-29 Set the release manager



Figure 7-30 Confirmation release one creation

- When selecting IT release 1, the validation activity can now be created from the **Activities** tab by using the Plus (+) icon and selecting **New Validation Activity** (Figure 7-31). Ivan, the ODM developer, creates and runs the test for the first time and Barbara, the business rule author, will approve the result, Figure 7-32 on page 100.

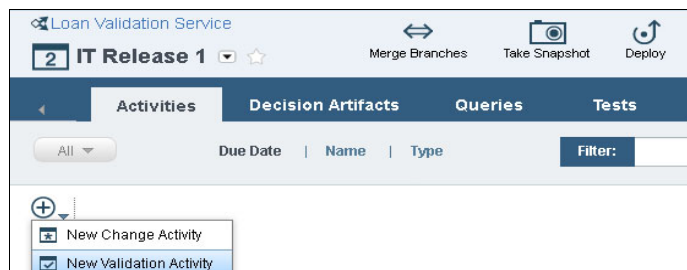


Figure 7-31 New Validation Activity

Figure 7-32 IT Release 1 Validation Activity creation

7.3.3 Validating IT Release 1

Validating IT Release 1 involves the following steps:

1. Ivan is the tester of the Initial IT release Validation Activity. When logged in to Decision Center, the **Work** tab shows that he must work on it, as shown in Figure 7-33.

Figure 7-33 Ivan's Work in Initial IT release

2. The validation activity is opened by clicking it. It consists of three tabs, as shown in Figure 7-34 on page 101:
 - The Test Plans tab is used to document the test plan associated with the validation activity
 - The Test Suites tab is used to test decision service operations against scenarios listed in Excel sheets.
 - The Simulations tab is used to determine how changes to rules or data affect the results of your business rule application before deployment. This process is not described further in this book.

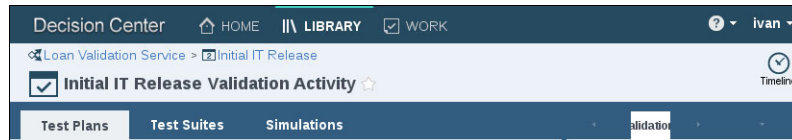


Figure 7-34 Initial IT Release Validation Activity Test Plans

- From the Test Suites tab, click on the **downward arrow** pointing to a tray (Figure 7-35) to generate an Excel Scenario file.

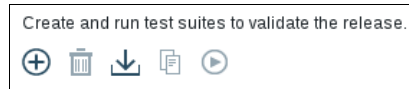


Figure 7-35 Test Suites action icons

- The Test Suite is an Excel spreadsheet and output parameter fields from the decision service operation to be checked and compared with expected values after execution are selected (Figure 7-36). In this example, only the Boolean **approved** field is selected.

Scenario files let you enter the scenario data and expected results. Generate and download the empty scenario file, define and enter scenario values using Excel, and add the completed scenario file to a test suite.

* **Filename:**

Scenario file format:

Locale:

Operation:
 loan validation

Select the tests to include in the scenario file.

* **Tests:**

Field	Operator
<input type="checkbox"/> the loan report	
<input type="checkbox"/> borrower	
<input type="checkbox"/> loan	
<input checked="" type="checkbox"/> approved	equals
<input type="checkbox"/> insurance	equals
<input type="checkbox"/> insurance rate	equals

Filter:

Figure 7-36 LoanValidationServiceTestSuite.xlsx generation (1)

5. An Excel file is generated and downloaded to be edited locally on a computer. It contains two tabs:
 - The Scenarios tab lists the scenarios and the input data to execute the scenario, as shown in Figure 7-37.

		the borrower							the loan			
Scenario ID	description	first name	last name	birth date	SSNCode	credit score	yearly income	zip code	start date	number of monthly payments	amount	Loan to Value
Scenario01		John	Doe	12/5/1968	123-45-6	600	100000	91320	6/1/2009	72	50000	0.7
Scenario02		John	Doe	12/5/1968	123-45-6	600	100000	91320	6/1/2009	72	80000	0.7

Figure 7-37 LoanValidationServiceTestSuite.xlsx generation (2)

- The expected results tab contains the expected results associated to each of the scenarios described in the Scenarios tab, as shown in Figure 7-38.

Scenario ID	the loan report is approved equals
Scenario01	TRUE
Scenario02	FALSE

Figure 7-38 LoanValidationServiceTestSuite.xlsx generation (3)

When the Excel Scenario file has been edited and filled in with the Scenarios and the Expected Results, it is possible to go to the next step, which is to create the Test Suites to execute this Scenario file.

6. From the Test Suites tab (Figure 7-35 on page 101) click the Plus sign (+) to create a Test Suite by first selecting the decision operation, as shown in Figure 7-39.

Select an operation

loan validation with score and gra...

☒ loan validation

loan validation

Used in production - input a loan request and a borrower - output the loan report. Rules from projects : Check, Determination and Scoring

Source Project:

Loan Validation Service

Extracted Rules:

Query: none
Validator: Default Validator

Ruleset Name:

loan_validation

Main Ruleflow:

loanvalidation

Input:

the borrower, the loan

Input/Output:

none

Output:

the loan report

Figure 7-39 Create Loan Validation Test Suite (1)

7. The Rule Execution Server running the test suite is selected in the Test Suite configuration screen, as shown in Figure 7-40.

The screenshot shows the 'New Test Suite' configuration screen in the Decision Center. The breadcrumb navigation is 'Loan Validation Service > Initial IT Release > Initial IT Release Validation Activity'. The page title is 'New Test Suite'. There are 'Save' and 'Save and Run' buttons in the top right. The configuration fields are as follows:

- Name:** 'Loan Validation Test Suite'
- Operation:** 'loan validation'
- Server:** 'Dev Decision Runner' (selected from a dropdown)
- Test rules from a snapshot:** ☐ (unchecked)
- Initial Snapshot:** 'Initial Snapshot' (selected from a dropdown)
- Decimal Precision:** 'Decimal places: 2' (selected from radio buttons)
- Scenarios:** (empty section)
- File format:** 'Excel 2007 (tabbed)' (selected from a dropdown)
- File to use:** 'InitITRelease- LoanValidationServiceTestSuite.xlsx' (with a 'Choose...' button and '(Update)' link)
- Report:**
 - Report name:** 'Report'
 - Expected execution details to include in report:**
 - ☐ The number of rules fired
 - ☐ The number of executed ruleflow tasks
 - ☐ The number of rules not fired
 - ☐ Create Excel file with report output values

Figure 7-40 Create Loan Validation Test Suite

Tip: Sometimes the Server drop-down list is empty or does not contain the server that you expect to see. This server must be defined in Decision Center by the administrator using the Enterprise Console. If defined, it might mean that the server is stopped or unreachable to the Decision Center. This must be solved by the ODM administrator, possibly with the help of the application server administrator and sometimes the network administrator.

8. Save and run the suite and verify that the expected results are returned, as shown in Figure 7-41.

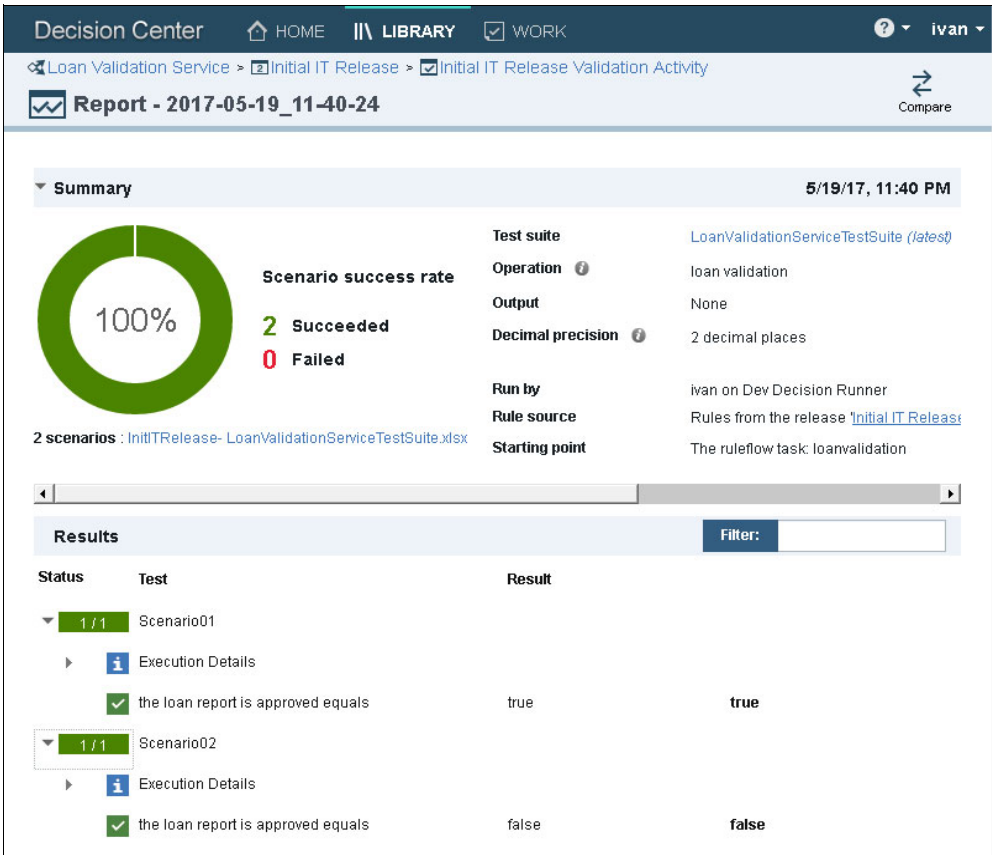


Figure 7-41 *LoanValidationServiceTestSuite* execution report

If the tests that are run within the IT release 1 Validation Activity return the expected results, the validation activity can be completed. This action also enables the release to be completed.

7.3.4 Completing IT Release 1

When every scenario of the test returns the expected results, the validation activity can be completed:

1. Ivan notifies that he has finished working with the validation activity, as shown in Figure 7-42.



Validation Activity	Stream	Validation Activity	Stream
 Created by adam May 17, 2017		 Created by adam May 17, 2017	
▼ Goals <i>The goals are not specified.</i>		▼ Goals <i>The goals are not specified.</i>	
Release Initial IT Release		Release Initial IT Release	
Decision Serv... Loan Validation Service		Decision Serv... Loan Validation Service	
Due Date May 30, 2017		Due Date May 30, 2017	
Status ⓘ In Progress		Status ⓘ In Progress	
Owner adam		Owner adam	
▼ Approvers		▼ Approvers	
barbara Not Review...		barbara Not Review...	
▼ Testers		▼ Testers	
ivan <div>Working Working Finish working</div>		ivan Finished ✓	

Figure 7-42 Validation Activity Finish working

2. Then Adam, the release manager, notifies the approver, Barbara, to **Proceed to approval**, as shown in Figure 7-43.

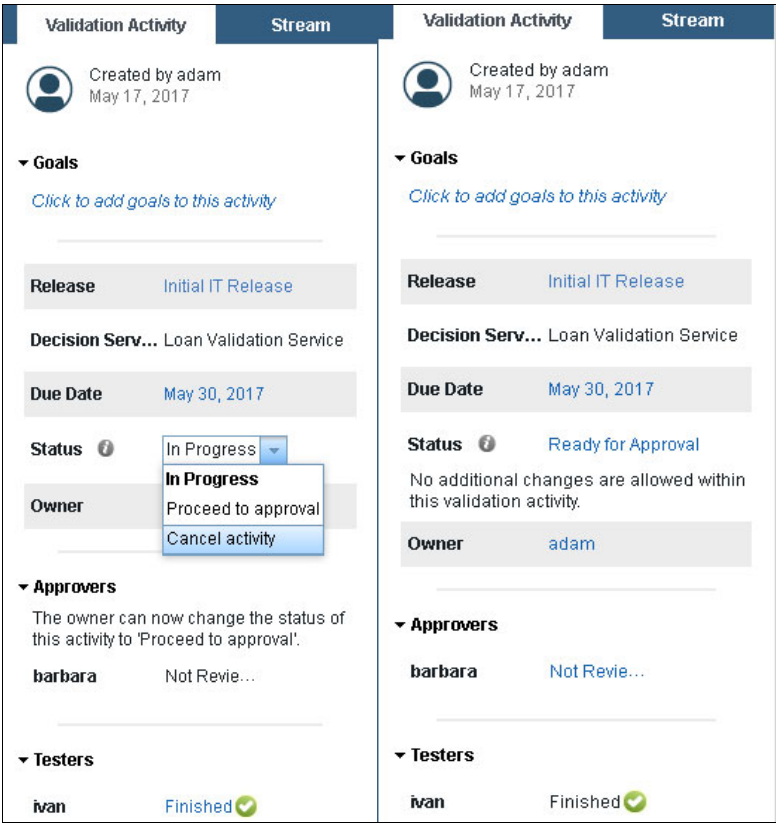


Figure 7-43 Validation Activity: Select Proceed to approval

- Barbara reviews the test suite test report and approves the changes. The validation activity status becomes Complete, as shown in Figure 7-44.



Validation Activity	Stream	Validation Activity	Stream
 Created by adam May 17, 2017		 Created by adam May 17, 2017	
▼ Goals <i>The goals are not specified.</i>		▼ Goals <i>The goals are not specified.</i>	
Release Initial IT Release		Release Initial IT Release	
Decision Serv... Loan Validation Service		Decision Serv... Loan Validation Service	
Due Date May 30, 2017		Due Date May 30, 2017	
Status ⓘ Ready for Approval No additional changes are allowed within this validation activity.		Status ⓘ Complete No additional changes are allowed within this validation activity.	
Owner adam		Owner adam	
▼ Approvers barbara Not Reviewed Not Reviewed Approve changes Reject changes		▼ Approvers barbara Approved ...	
▼ Testers ivan Finished ✓		▼ Testers ivan Finished ✓	

Figure 7-44 Validation Activity Approve changes

4. When all activities of the release are completed, Adam, the owner of the release changes the status to Proceed to approval, Figure 7-45.







Release	Stream
 Created by adam May 17, 2017	 Created by adam May 17, 2017
▼ Goals 	▼ Goals 
Initial IT Release to validate Initial release before starting Initial Business Release	Initial IT Release to validate Initial release before starting Initial Business Release
▶ Linked Projects	▶ Linked Projects
Parent Release Initial Release	Parent Release Initial Release
Decision Serv... Loan Validation Service	Decision Serv... Loan Validation Service
Due Date May 30, 2017	Due Date May 30, 2017
Status  <div>In Progress In Progress Proceed to approval Cancel release</div>	Status  Ready for Approval
Owner	Owner adam
▼ Approvers The owner can now change the status of this release to 'Proceed to approval'.	▼ Approvers
paul Not Review...	paul Not Review...

Figure 7-45 IT release Proceed to approval

- Paul, the approver of the validation activity, approves the changes, as shown in Figure 7-46.

The figure displays two side-by-side screenshots of a software release management interface, showing the transition of a release from 'Ready for Approval' to 'Complete'.

Left Screenshot (Ready for Approval):

- Release:** Created by adam, May 17, 2017.
- Goals:** Initial IT Release to validate Initial release before starting Initial Business Release.
- Linked Projects:** (Empty list)
- Parent Release:** Initial Release
- Decision Serv...:** Loan Validation Service
- Due Date:** May 30, 2017
- Status:** Ready for Approval. Info icon. No additional changes are allowed within this change activity.
- Owner:** adam
- Approvers:**
 - paul: Not Reviewed, Not Reviewed, **Approve changes** (highlighted), Reject changes

Right Screenshot (Complete):

- Release:** Created by adam, May 17, 2017.
- Goals:** Initial IT Release to validate Initial release before starting Initial Business Release.
- Linked Projects:** (Empty list)
- Parent Release:** Initial Release
- Decision Serv...:** Loan Validation Service
- Due Date:** May 30, 2017
- Status:** Complete. Info icon. No additional changes are allowed within this change activity.
- Owner:** adam
- Approvers:**
 - paul: Approved ...

Figure 7-46 IT release Approve changes

The IT release 1 is now in state *Complete*. Business users can now finalize the implementation Loan Validation Service.

7.3.5 Creating Business Release 1

Now that the first IT release is validated, Paul, the release manager, can create a first business release, as described in “Business releases” on page 92, and a change activity for the Rule Authors to create and edit the necessary rules:

1. From Decision Center Library tab, Paul clicks the Plus sign (+) or the arrow next to Initial Release and then sets Rachel, the Regional Manager, as approver of the release, as shown in Figure 7-47.

Figure 7-47 Business release 1 creation

- ## Create a Change Activity

Give your activity a name:

Specify goals for this new activity:

B I U |
 |
 |
 |

development of the business rules based on the Initial IT Release of the Loan Validation Service

Specify a due date:

Specify who will participate in this activity:

Owner:

Approvers:

Authors:

Change Activity

Stream

Created by paul
May 20, 2017

Goals

 - development of the business rules based on the Initial IT Release of the Loan Validation Service

Release	Business Release 1
Decision Serv...	Loan Validation Service
Due Date	Jun 3, 2017
Status	In Progress
Owner	paul

Approvers

 - rachel Not Reviewing

Authors

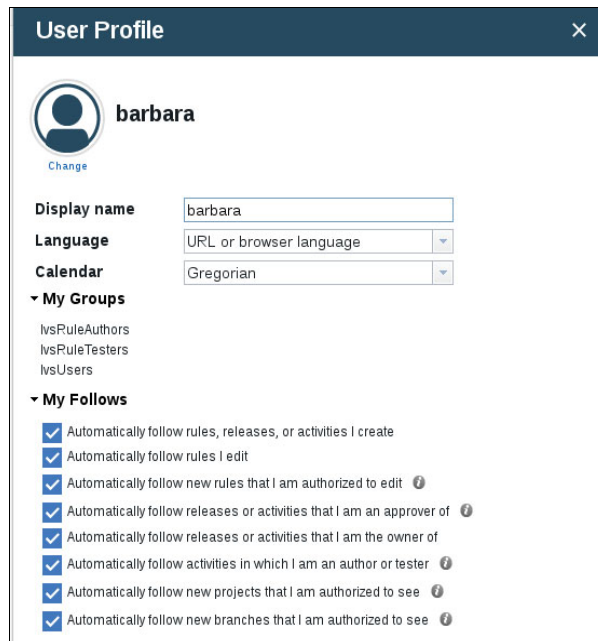
 - barbara Working

Barbara, the rule author, can now start working on the Loan Validation Service.

7.3.6 Changing Business release 1

Changing Business release 1 involves the following steps:

1. When Barbara first connects to the Business Console, it is recommended that she review and adjust her profile by clicking her name on the upper right side of the browser page. A configuration window displays, as shown in Figure 7-49.



The screenshot shows a 'User Profile' window with a dark blue header and a close button (X) in the top right corner. Below the header is a profile section for 'barbara' with a circular profile picture icon and a 'Change' link. The main content area contains several settings:

- Display name:** A text input field containing 'barbara'.
- Language:** A dropdown menu showing 'URL or browser language'.
- Calendar:** A dropdown menu showing 'Gregorian'.
- My Groups:** A section with a downward arrow icon, listing three groups: 'lvsRuleAuthors', 'lvsRuleTesters', and 'lvsUsers'.
- My Follows:** A section with a downward arrow icon, containing eight checkboxes, all of which are checked:
 - Automatically follow rules, releases, or activities I create
 - Automatically follow rules I edit
 - Automatically follow new rules that I am authorized to edit (with an info icon)
 - Automatically follow releases or activities that I am an approver of (with an info icon)
 - Automatically follow releases or activities that I am the owner of
 - Automatically follow activities in which I am an author or tester (with an info icon)
 - Automatically follow new projects that I am authorized to see (with an info icon)
 - Automatically follow new branches that I am authorized to see (with an info icon)

Figure 7-49 User Profile

Display Name, Language, and Calendar enable control of the Business Console display options.

My Groups lists the groups the user is part of and helps understand which decision service projects are accessible by the user and with which role. In Figure 7-49 on page 112 Barbara is part of the Loan Validation Service (lvs) users, rule testers, and rule authors. She can be any of the following roles:

- Owner or approver of releases and activities
- Rule author of change activities
- Tester of validation activities

Her author role limits her to action rules and decision table edition as seen in Chapter 4, “Securing the Decision Center” on page 31.

My Follows is used to subscribe automatically to Business Console events the user is involved with.

Tip: Also, hover over a release, Activity, Folder, or Individual rule and select the star sign to subscribe to follow changes affecting an individual artifact.

2. In the *Work* tab, Barbara also sees that she must work on the business release 1 activity release.

3. The business release 1 activity release displays the banner shown in Figure 7-50.

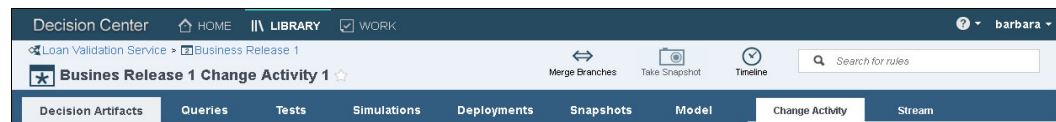


Figure 7-50 Change activity banner

4. Rule authors mainly work with decision artifacts to browse the decision service project artifacts, create new ones and modify existing ones, as shown in Figure 7-51.

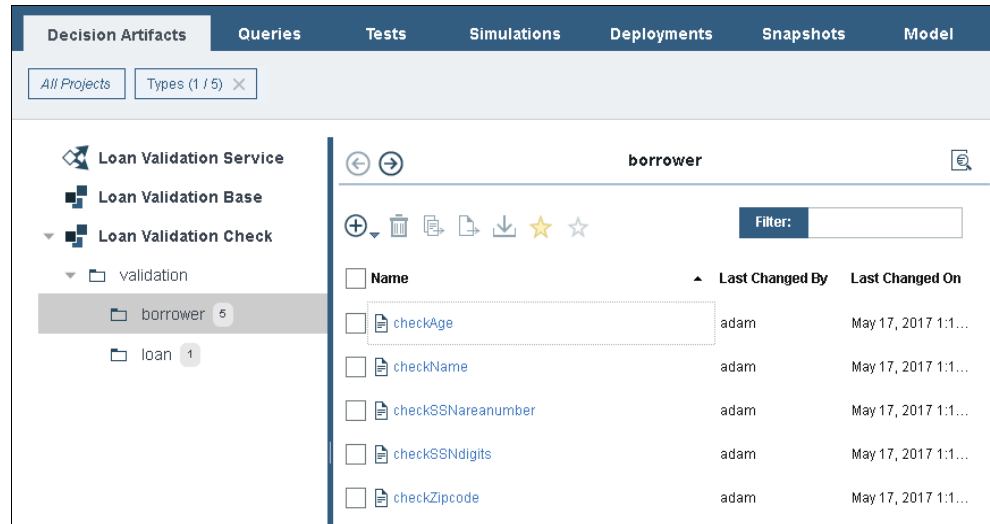


Figure 7-51 Decision Artifacts tab

The Business Console rule editor assists the user when editing action rules, in this case with simple `if... then... else` code, as shown in Figure 7-52.

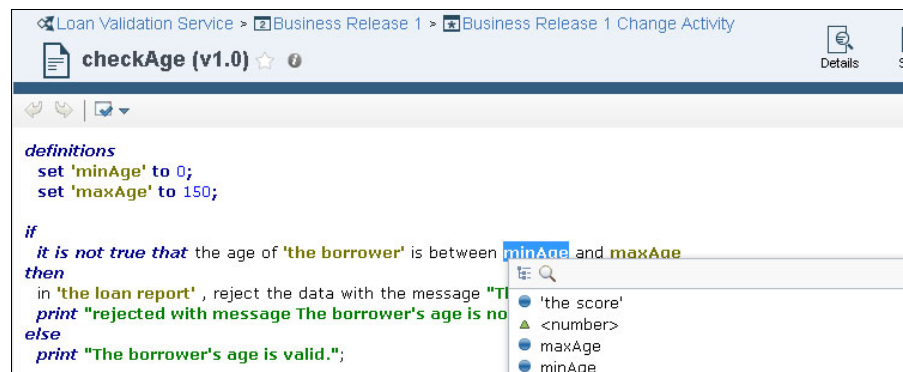
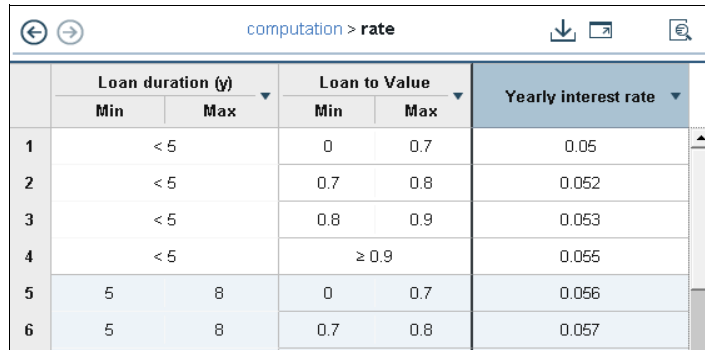


Figure 7-52 Business Console Action Rule Editor

The Business Console rule editor can also assist the user with an Or Decision table, such as that shown in Figure 7-53.



computation > rate					
	Loan duration (y)		Loan to Value		Yearly interest rate
	Min	Max	Min	Max	
1	< 5		0	0.7	0.05
2	< 5		0.7	0.8	0.052
3	< 5		0.8	0.9	0.053
4	< 5		≥ 0.9		0.055
5	5	8	0	0.7	0.056
6	5	8	0.7	0.8	0.057

Figure 7-53 Business Console Decision Table

The *Tests* and *Simulations* tab, as in the case of the validation activity, are used to execute test scenarios from Excel sheets.

5. When Barbara is done with the rule development and tests are successful, the change activity is declared finished.

A validation activity can be created, as in the case of the IT release, to formalize the release validation.

7.3.7 Business Release 1 completion

To complete the Business Release 1, every change activity and then validation activity must be complete. In the sample scenario, only one change activity has been created:

1. Barbara first declares her work **Finish working** on the IT release 1 change activity, as shown in Figure 7-54.



Change Activity	Stream	Change Activity	Stream
 Created by paul May 20, 2017		 Created by paul May 20, 2017	
▼ Goals development of the business rules based on the Initial IT Release of the Loan Validation Service		▼ Goals development of the business rules based on the Initial IT Release of the Loan Validation Service	
Release Business Release 1		Release Business Release 1	
Decision Service Loan Validation Service		Decision Service Loan Validation Service	
Due Date Jun 3, 2017		Due Date Jun 3, 2017	
Status ⓘ In Progress		Status ⓘ In Progress	
Owner paul		Owner paul	
▼ Approvers rachel Not Reviewed		▼ Approvers rachel Not Reviewed	
▼ Authors barbara <div>Working Working Finish working</div>		▼ Authors barbara Finished ✓	

Figure 7-54 Change activity Finish working

- Then Paul, the release manager, sets the change activity status to **Proceed to approval** (Figure 7-55).









Change Activity	Stream
<p> Created by paul May 20, 2017</p> <p>▼ Goals </p> <p>development of the business rules based on the Initial IT Release of the Loan Validation Service</p> <hr/> <p>Release Business Release 1</p> <p>Decision Service Loan Validation Service</p> <p>Due Date Jun 3, 2017</p> <p>Status  In Progress ▼</p> <p>Owner </p> <p>▼ Approvers</p> <p>The owner can now change the status of this activity to 'Proceed to approval'.</p> <p>rachel Not Reviewed</p> <hr/> <p>▼ Authors</p> <p>barbara Finished </p>	<p> Created by paul May 20, 2017</p> <p>▼ Goals </p> <p>development of the business rules based on the Initial IT Release of the Loan Validation Service</p> <hr/> <p>Release Business Release 1</p> <p>Decision Service Loan Validation Service</p> <p>Due Date Jun 3, 2017</p> <p>Status  Ready for Approval</p> <p>No additional changes are allowed within this change activity.</p> <p>Owner paul</p> <hr/> <p>▼ Approvers</p> <p>rachel Not Reviewed</p> <hr/> <p>▼ Authors</p> <p>barbara Finished </p>

Figure 7-55 Change activity Proceed to approval

3. Finally, Rachel, the Regional Manager and approver of Business Release 1 Change Activity 1, selects **Approve changes** and this automatically sets the status to *Complete*, as shown in Figure 7-56.

Change Activity	Stream
<p>Created by paul May 20, 2017</p> <p>▼ Goals development of the business rules based on the Initial IT Release of the Loan Validation Service</p> <p>Release Business Release 1</p> <p>Decision Service Loan Validation Service</p> <p>Due Date Jun 3, 2017</p> <p>Status ⓘ Ready for Approval No additional changes are allowed within this change activity.</p> <p>Owner paul</p> <p>▼ Approvers rachel ▼ Authors barbara</p>	<p>Created by paul May 20, 2017</p> <p>▼ Goals development of the business rules based on the Initial IT Release of the Loan Validation Service</p> <p>Release Business Release 1</p> <p>Decision Service Loan Validation Service</p> <p>Due Date Jun 3, 2017</p> <p>Status ⓘ Complete No additional changes are allowed within this change activity.</p> <p>Owner paul</p> <p>▼ Approvers rachel ▼ Authors barbara</p>

Figure 7-56 Change activity Approve changes

The business release can now be completed, finishing every other change and validation activity, as illustrated in “Completing IT Release 1” on page 105.

Repeat for each release.

7.4 Conclusion

This chapter explained how the decision governance framework is used within Business Console.

It detailed the underlying concepts of release, change activity, and validation activity. Their lifecycles and the overall workflow was explained step by step.

The notion of IT and Business releases were added to the decision governance framework to best coordinate short-term business rule update requirement and long-term IT projects necessary to improve or extend the initial decision service project organization and data model.



Deployment

Deployment is the activity of creating a RuleApp and making it available to a Rule Execution Server (RES). Deployment can be applied through Decision Center, Rule Designer, and the Rule Execution Server management console.

This chapter covers the following topics:

- ▶ Choosing a deployment strategy
- ▶ Difference between production and non-production deployment
- ▶ Versioning policies
- ▶ Managing resources
- ▶ Conclusion

Terminology used:

- ▶ **RuleApp** identifies the decision service name. It contains one or more rulesets.
- ▶ **Ruleset** identifies a decision service operation.
- ▶ **Decision operation** includes all the settings needed to define the contents of a ruleset and its parameters.
- ▶ **Deployment configuration** defines how the decision service will be deployed as a RuleApp to Rule Execution Server.
- ▶ **Rule Execution Server** provides the runtime environment for running and monitoring RuleApps.
- ▶ **Business Console** is the preferred UI environment for business users to deploy decision services.
- ▶ **Enterprise Console** is the environment for advanced deployment administration.

8.1 Choosing a deployment strategy

Decision Center provides great flexibility in where, when, and who should deploy a decision service. This section provides guidelines and best practices for deploying RuleApps.

8.1.1 IT Centric deployments

For IT Centric changes, rigorous testing is required and the RuleApp should be deployed through a series of staging environments before production. For example, deploy in the System Integration Test environment, User Acceptance Test environment, and Preproduction environment. Only after the RuleApp has been fully tested on these environments should it be deployed to Production. IT Centric deployments are usually performed through Rule Designer, as shown in Figure 8-1.

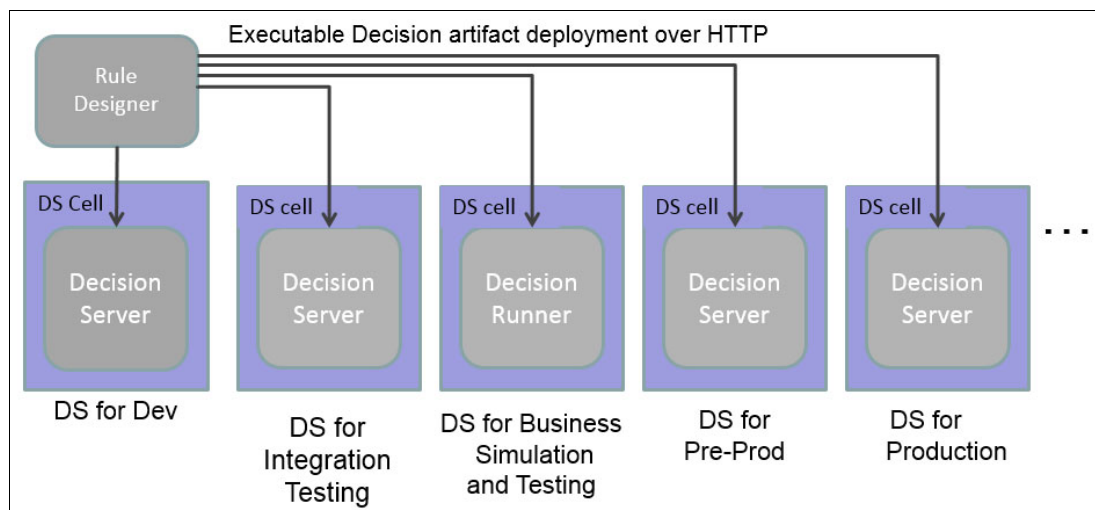


Figure 8-1 IT Centric Deployments

8.1.2 Business Centric deployments

For Business Centric changes, the decision service should have already been verified by the decision governance process, so it can bypass the staging environments and be deployed directly to Production. Any business changes not verified by the decision governance process should go through the staging environments first (dotted line). See Figure 8-2 on page 121.

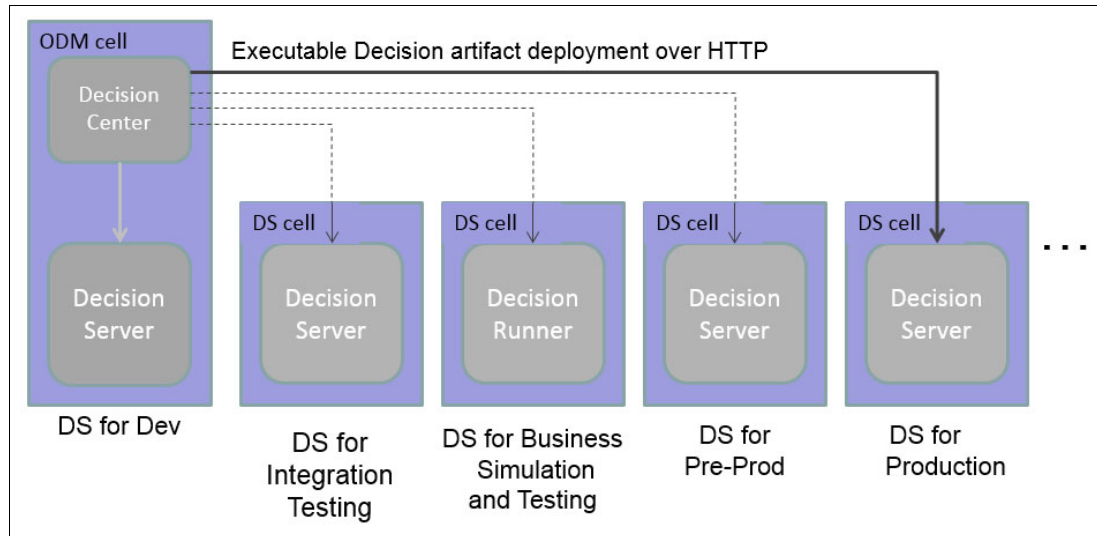


Figure 8-2 Business Centric Direct Deployment

8.1.3 Two-phase deployment

In some cases, there is a firewall between Production and Business environments, so there is no possibility of direct deployment. Instead, a RuleApp is saved to a shared folder and then the deployment manager deploys it using the RES console. This is known as *two-phase deployment*:

1. Business users generate the RuleApp from the Business Console and save it in a shared location (the staging area) for the deployment manager to pick up.
2. At an agreed time, the deployment manager determines the correct versioning policy and deploys the RuleApp to Production. Deployment can be automated by using a CRON job or some other production deployment tooling.

Two-phase deployment decouples the roles of Decision Center users who generate the RuleApp archive, from the deployment manager who manages the runtime environments. It prevents Decision Center users from inadvertently deploying a RuleApp at the wrong time. For example, an approved business release containing a seasonal promotion could be inadvertently deployed two weeks too early, causing the wrong rules to be fired. The deployment manager acts as the safeguard.

Figure 8-3 shows a two-phase deployment for a business release. This approach also applies to IT Centric releases. You can see in the diagram that it is only the *Production* that is two phases. The staging environments are deployed directly from Decision Center, but you could also choose to do them in two phases.

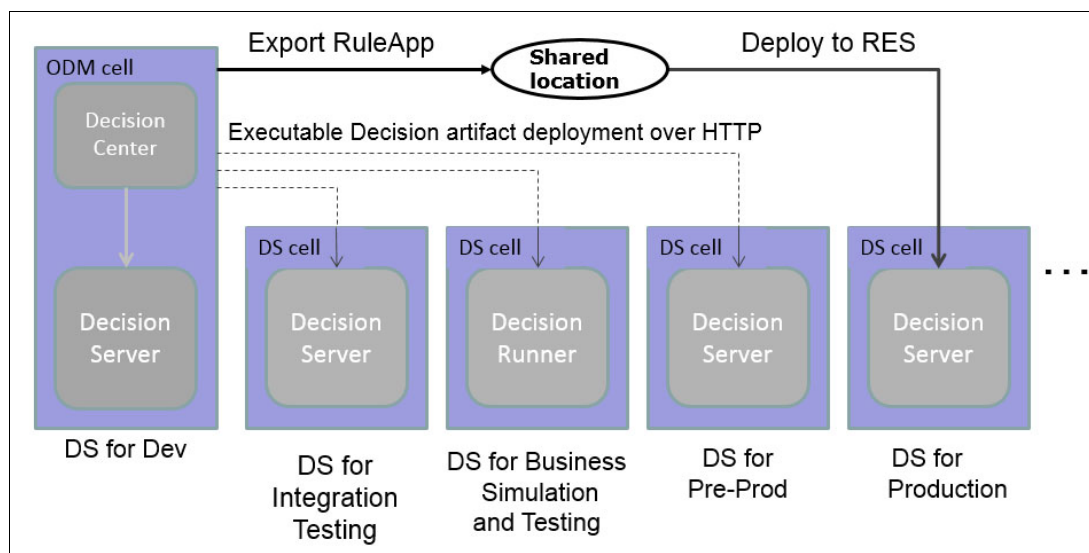


Figure 8-3 Two-phase deployment

The disadvantage of two-phase deployment is that the RuleApp might have associated resources that must be managed, 8.4, “Managing resources” on page 124.

8.2 Difference between production and non-production deployment

When you define a deployment configuration, you choose either production or non-production.

Attention: When choosing between production and non-production servers, Decision Center does not check that the selected server has been installed with a production license.

Selecting the deployment server type restricts deployment to specific servers:

- **Non-production:** This setting imposes restrictions on deployment from the Business Console within the governance framework. In a change activity, you can deploy only to non-production servers.
- **Production:** Deployment to production is available after a decision service release is complete in the governance framework.

When working within the governance framework, a deployment configuration can only be created or edited within a change activity.

Example: Configuring deployment to production and non-production

Any user from the `rtsConfigManagers` group can create a deployment configuration (in this case, users Adam and Terry).

To create a deployment configuration, complete the following steps:

1. Select **Loan Validation Service** from the Library tab of the Business Console.
2. Select the **In Progress** Spring release.
3. Select an **In Progress** activity.
4. On the Deployments tab, create a new deployment configuration using the **New (+)** button. The deployment configuration creation page opens.
5. On the General tab, select the **configuration type**. The type that you choose depends on the target server that you select on the Targets tab. On the Groups tab, you can select which groups of users are allowed to use this deployment configuration. This way, you can limit the deployment to specific users:
 - If the target server is the development server, select **non-production** as the **configuration type**, and on the Groups tab, associate the `1vsUsers` group to allow any business user to deploy the Loan Validation Service to the development Rule Execution Server.
 - If the associated target server is the production server, select **Production** as the **configuration type**, and on the Groups tab, associate the `dcAdministrators` group to restrict the deployment to the production Rule Execution Server to be performed only by the `rtsAdministrator`.

8.3 Versioning policies

At run time, client applications distinguish between RuleApp and ruleset versions by using a versioning policy. In this section, we provide best practices for implementing this policy.

8.3.1 RuleApp and ruleset versioning policy

A RuleApp is the container for one or more rulesets representing decision services.

Important: A change to a RuleApp version represents a change to the BOM, whereas a change to a ruleset version represents a change to a decision service.

A deployed RuleApp is identified by using a ruleset path as follows:

RuleApp/RuleApp_major.RuleApp_minor/Ruleset/Ruleset_major.Ruleset_minor

In this example, use the following guidelines:

- | | |
|----------------------|---|
| RuleApp_major | Change the RuleApp major version if there is a BOM/XOM change that affects several decision service signatures within the RuleApp. |
| RuleApp_minor | Change the minor RuleApp version if there is a BOM/XOM change that does not affect any decision service signatures, such as a Java code change within a method. |

Ruleset_major	Change the ruleset major version if the decision service signature has changed and caused incompatibility between the previous release.
Ruleset_minor	Change the minor ruleset version if the decision service signature has not changed and the change is compatible with the previous release. This is typically a business release where a change to business policy has no effect on the decision service signature.

For more information on versioning, see [IBM Knowledge Center](#).

8.3.2 Running against multiple deployed versions

When multiple versions of RuleApps are deployed to Rule Execution Server, a versioning scheme can be adopted to pin clients to a specific major version but still allow them to accept minor version updates.

Example: Loan Validation versioning

The initial deployment version of the Loan Validation services is:

```
LoanValidationRuleApp/1.0/LoanValidationProductionOperation/1.0
```

The client specifies a ruleset path that allows minor BOM/XOM and decision service increments using a wildcard (*) character:

```
LoanValidationRuleApp/1.*/LoanValidationProductionOperation/1.*
```

A new minor ruleset version is deployed a week later:

```
LoanValidationRuleApp/1.0/LoanValidationProductionOperation/1.1
```

The path is accepted by the client and the new RuleApp is run. A month later a minor BOM release is issued:

```
LoanValidationRuleApp/1.1/LoanValidationProductionOperation/1.1
```

The path is accepted by the client and the new RuleApp is run. Six months later a major RuleApp version is deployed:

```
LoanValidationRuleApp/2.0/LoanValidationProductionOperation/2.0
```

The client does not select this update because it is a major release and it would cause compatibility issues.

8.4 Managing resources

When you manually deploy a new version of a decision service from the RES console, you must ensure that resources (XOM and libraries) needed by the decision service are correctly deployed and linked.

Cases to consider:

- ▶ If you deploy a new business release that does not contain any XOM update, you must ensure that the new release links to the existing XOM.
- ▶ If you deploy a release that contains a XOM change, you need to handle the linkage and deployment of the updated resources.

- If you work in an IT Centric approach, the deployment manager deploys XOM updates from Rule Designer to Rule Execution Server. If you work in a Business Centric approach, the deployment manager deploys the updated XOM to Rule Execution Server from the Resources folder in Business Console.

Tip: In Rule Designer, ensure that the **Deploy the XOM** option is set to **Yes** on the Overview tab of the deployment configuration. This instructs Rule Designer to deploy the managed Java XOM at the same time as the decision service RuleApp.

We suggest that the updated XOM is deployed to the Rule Execution Servers before the first synchronization with Decision Center.

Manual Deployments with XOM change:

To manually deploy, complete the following steps:

1. In Rule Designer, select **Deploy the XOM** → **Yes** (see 8.4.1, “Managed Java XOM”).
2. In Rule Designer, at the decision service level in the properties, select **Enable XOM management in Decision Center** (see 8.4.2, “Enabling XOM management in Decision Center”). This ensures that the XOM and resources will be exported from Rule Designer to Decision Center in the Resources folder.
3. From Business Console, the business user generates the RuleApp.
4. The deployment manager then performs the following steps:
 - a. Retrieves the RuleApp from the shared location.
 - b. Gets the resources from the Business Console Resources folder.
 - c. Deploys the RuleApp and associated resources with ANT scripts or REST APIs.

8.4.1 Managed Java XOM

Within Rule Designer, select **Deploy the XOM** → **Yes** in the deployment configuration Overview tab to enable XOM management, which ensures that Rule Designer automatically manages the `ruleset.managedxom.uris` property and computes its value.

Note: `Ruleset.managedxom.uris` is a ruleset property, with its own section in the RES console: it is not displayed in the list of ruleset properties under **Show ruleset properties** of a ruleset, but in the “Show Managed URIs” .

For more information, see the *Managed Java XOM* topic in [IBM Knowledge Center](#).

8.4.2 Enabling XOM management in Decision Center

Tip: We suggest that you **Enable XOM management in Decision Center**. This property ensures that when synchronizing from Rule Designer to Decision Center, the XOM project is published as well. This property is set in the properties of the decision service, under the decision service item.

Selecting the **Enable XOM management in Decision Center** property in Rule Designer ensures that when you synchronize your decision service from Rule Designer to Decision Center, the XOM is published to the Resources folder. The Resources folder is found in the `resources/xom-libraries` directory. All RuleApps created from this decision service reference this version of the XOM. At deployment time, Decision Center determines whether this version of the XOM is already present in Rule Execution Server. If not, Decision Center deploys the XOM along with the RuleApp.

The deployment manager is able to pick up the appropriate resources from the Resources folder and link them to the RuleApp for deployment by using ANT scripts, using the RES REST API, or manually.

For more information, see the *XOM Deployment from Decision Center* topic in [IBM Knowledge Center](#).

8.5 Conclusion

This chapter explained the different deployment approaches for deploying a decision service:

- ▶ IT Centric deployment, usually performed from Rule Designer
- ▶ Business Centric deployment, performed by business users from the Business Console
- ▶ Two-phase deployment that can be used when Decision Center and Rule Execution Server are disconnected

This chapter also explained the difference between non-production and production deployment. Finally, it gave suggestions regarding versioning policies and managing resources to help you build your own deployment strategy.



ODM DevOps

This chapter defines DevOps as the automated build, test, and deployment of decision services. It discusses two different patterns:

- ▶ IT Centric
- ▶ Business Centric

The reader might want to combine parts of each pattern to come up with their own design.

This chapter then discusses ways to implement these patterns using IBM UrbanCode™ Deploy and ODM APIs. The following topics are covered:

- ▶ IT Centric DevOps
- ▶ Business Centric DevOps
- ▶ DevOps orchestration with UrbanCode Deploy
- ▶ Suggested APIs for DevOps
- ▶ Conclusion

The following terminology is used in this chapter:

RuleApp	The <i>rule application archive</i> containing runnable rule artifacts
Commit	To check in code or rules to the source code repository
Publish	To push rule artifacts from Rule Designer to Decision Center
Update	To pull rule artifacts from Decision Center to Rule Designer
Deploy	The deployment of a decision service for execution
Source Code Repository	An archive storing source code and business rules
ODM DevOps API	The application programming interface to build and deploy RuleApps

9.1 IT Centric DevOps

The IT Centric DevOps pattern builds and deploys decision services in a pure development environment. The source code repository is the “golden” source of all artifacts, including business rules. The business can view but not change rules. If the business requires a change, it submits a request via the agreed change process, where it is assigned to a developer.

The developer applies and commits the change to the source code repository. DevOps builds the RuleApp and applies a series of automated and manual tests to progress the RuleApp through staging environments to preproduction. Finally, the RuleApp is deployed to Production at an agreed-upon time and day.

9.1.1 Advantages

The advantage of this approach is that all builds are contained within the IT governance domain. There is no parallel activity between business and IT. DevOps is a purely technical process in the hands of IT.

Both technical and business changes can be made in each build, so there is no distinction between business releases and technical releases.

9.1.2 Disadvantages

The disadvantage is that Decision Center is not used for rule changes, so it is not possible to deliver true business agility. Rule changes must go through the full IT build and test cycle, which could take months. This resource cost is likely too high for simple rule changes.

9.1.3 IT Centric DevOps sequence

IT Centric DevOps is initiated at the start of the initial release, or after the business has requested a change to the production release. The following sequence indicates whether the step is manual [M] or automatically scripted [A]. Step 5 can be omitted for technical changes not requiring business rule changes.

The Business Policy Change scenario includes the following steps:

1. [M] Rule Developer makes a Business change.
2. [M] Rule Developer unit tests the change.
3. [M] Rule Developer commits the change to the source code repository.
4. [A] Build Server triggers a RuleApp build.
5. [A] The RuleApp is deployed to the Rule Execution Server on the Test platform.
6. [A] The release is regression tested.
7. [A] The change is published to Decision Center.
8. [M] The business reviews the changes.
9. [A] If the business approves, the release is deployed to System Integration Test (SIT), User Acceptance Test (UAT), and Pre-Production.

10.[M] Acceptance tests are performed in each environment.

11.[A] If acceptance tests pass, the release is deployed to Production at an agreed-upon time and day.

This sequence is shown diagrammatically in Figure 9-1.

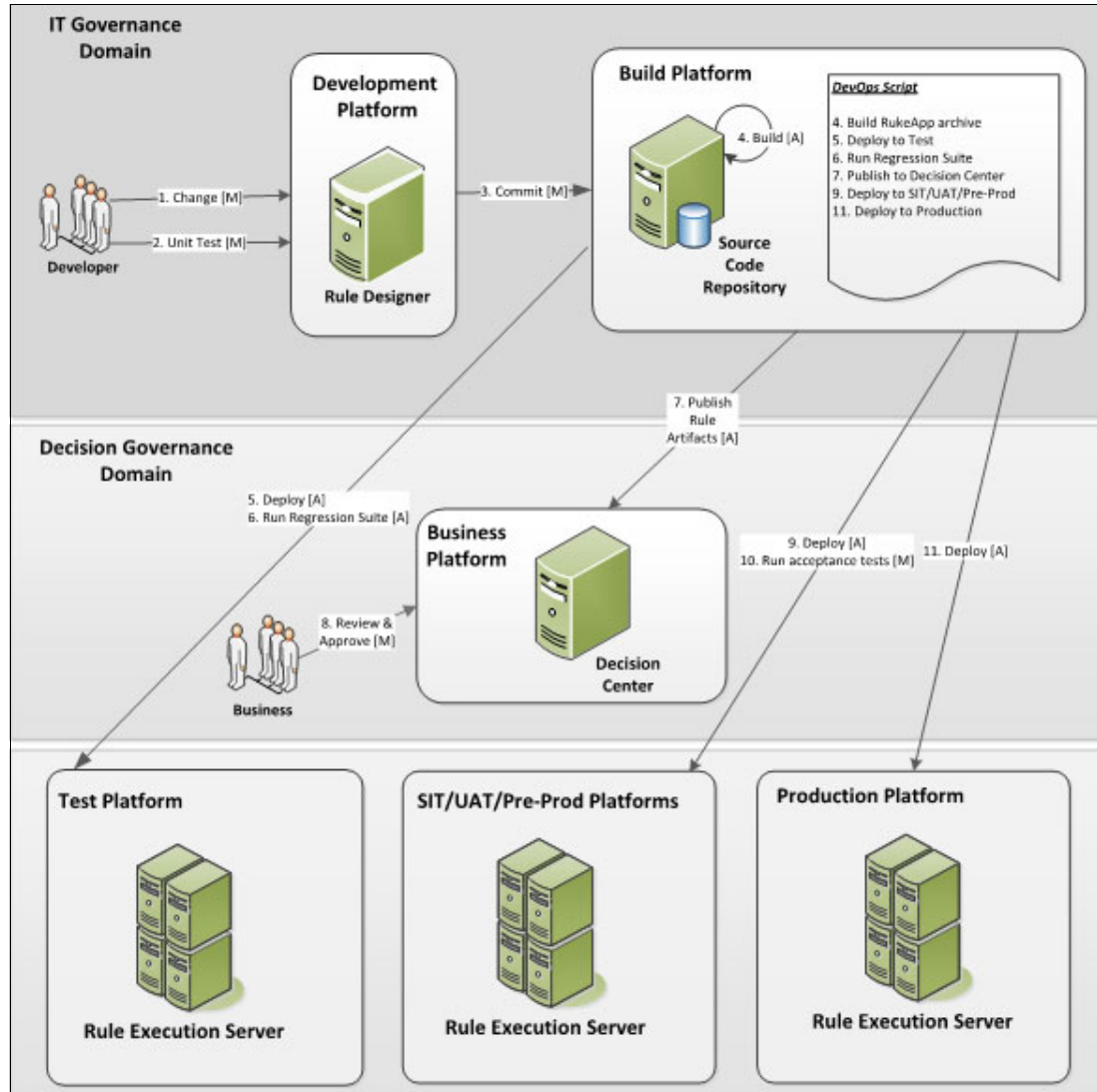


Figure 9-1 IT Centric DevOps

9.2 Business Centric DevOps

In this pattern, Decision Center is the “golden” source of rules.

Business Centric DevOps requires rule artifacts to be changed and approved in Decision Center using decision governance. Scripts automate the build and deployment of RuleApps from Decision Center. Technical changes are applied by a separate IT build process using Rule Designer. Technical changes require synchronizing to Decision Center.

9.2.1 Advantages

The advantage of Business Centric DevOps is that you get true business agility using decision governance.

9.2.2 Disadvantages

The disadvantage is that whenever there is a technical change, such as a change to the business object model, a separate build and synchronization process is required. If technical changes are infrequent this is not a problem.

9.2.3 Regular cycle: Business change

In this sequence, we assume that the initial release of the decision service has already been built and published to Decision Center, as described in the IT Centric DevOps.

The DevOps process is initiated after the business has completed making changes in Decision Center and the release has been approved by the business owner. The sequence is shown diagrammatically in Figure 9-2 on page 131. Step 8 can be omitted for most business rule changes.

Automatically scripted steps are indicated by an [A], and manual steps are indicated by [M]:

1. [M] Business creates and approves a release in Decision Center.
2. [A] The release approval is detected¹, triggering a build on the Build Server.
3. [A] The Build Server uses Decision Center APIs to build a RuleApp from the approved release.
4. [A] The Decision Center APIs automatically run and verify business tests. This step is only required if the decision governance framework is not enforcing validation activities.
5. [A] The RuleApp is deployed to Rule Execution Server on the Test platform.
6. [A] The release is regression tested
7. [A] The release is deployed to System Integration Test (SIT), User Acceptance Test (UAT), and Pre-Production.
8. [M] Acceptance tests are performed in each environment
9. [A] If acceptance tests pass, the release is deployed to Production at an agreed time and day.

¹ Use the custom session controller to detect a state transition from Ready for Approval to Complete on the release. See 9.4, "Suggested APIs for DevOps" on page 133.

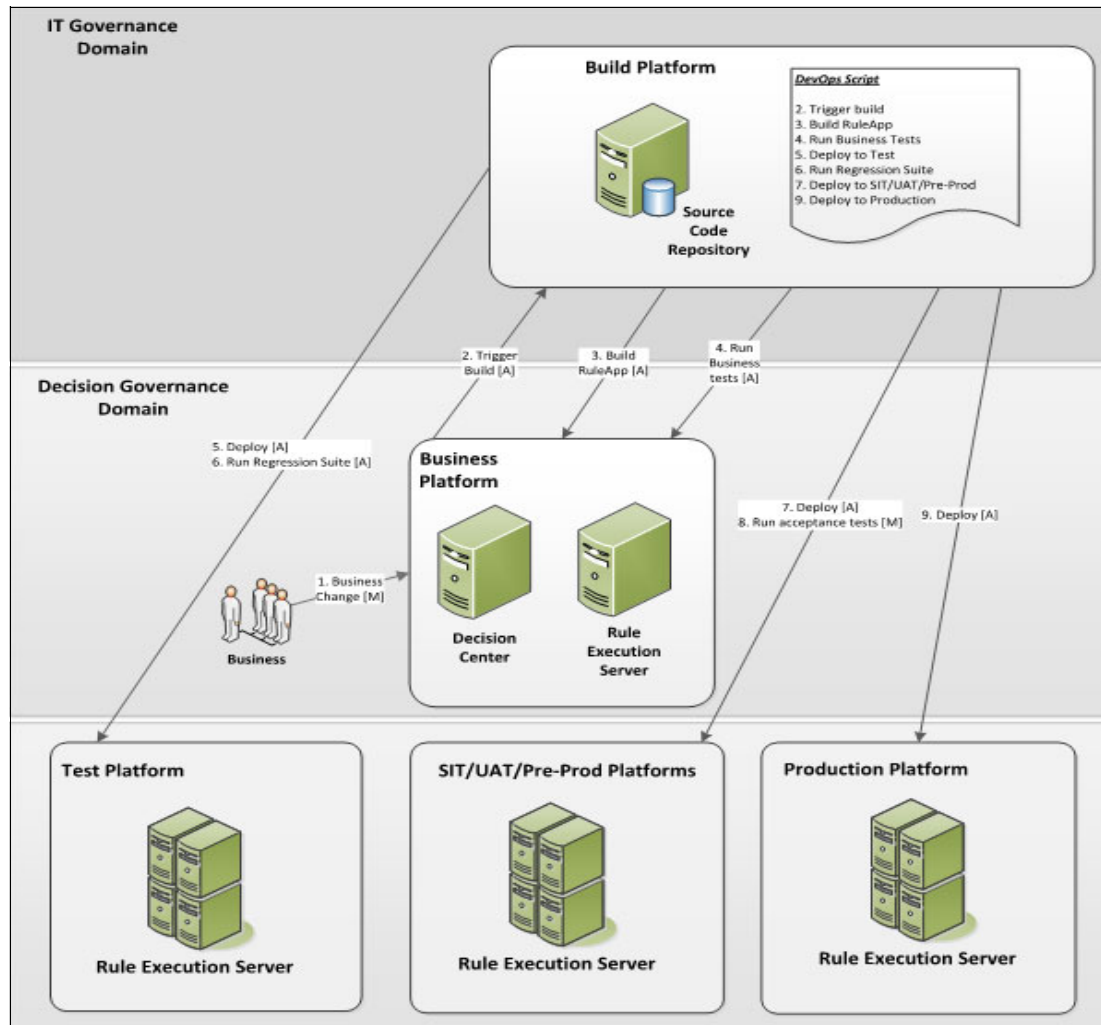


Figure 9-2 Business Centric DevOps: Business change

9.2.4 Irregular cycle: Technical change

This sequence begins when the business requests a change that it cannot make itself. This change is a technical change, and is assigned to a developer where it is governed within the IT domain. Technical changes to the XOM are outside the scope of Decision Center, and are committed to the IT source code repository. All other technical changes, such as changes to the BOM, are applied by the Rule Developer in Rule Designer, but published to Decision Center as part of the technical release, as shown in Figure 9-3 on page 132.

In a situation where both technical and business changes are required for a single release, we suggest two separate DevOps cycles, one to build and test the technical changes and a second to build, test, and deploy the business changes.

Automatically scripted steps are indicated by an [A] and manual steps are indicated by [M]:

1. [M] Developer synchronizes (updates) changes from Decision Center to their development platform.
2. [M] Developer makes technical change.
3. [M] Developer unit tests change.

4. [M] Developer commits XOM changes to source code control.
5. [M] Developer publishes technical changes to Decision Center.
6. [A] RuleApp built using Decision Center API.
7. [A] Business scenarios in Decision Center are run and verified.
8. [A] The RuleApp is deployed to Rule Execution Server on the Test Platform.
9. [A] The release is regression tested.
- 10.[A] The release is deployed to System Integration Test (SIT), User Acceptance Test (UAT) and Pre-Production.
- 11.[M] Acceptance tests are performed in each environment.
- 12.[A] If acceptance tests pass the release then needs to be deployed to Production at an agreed-upon time and day.

Figure 9-3 shows an Irregular cycle.

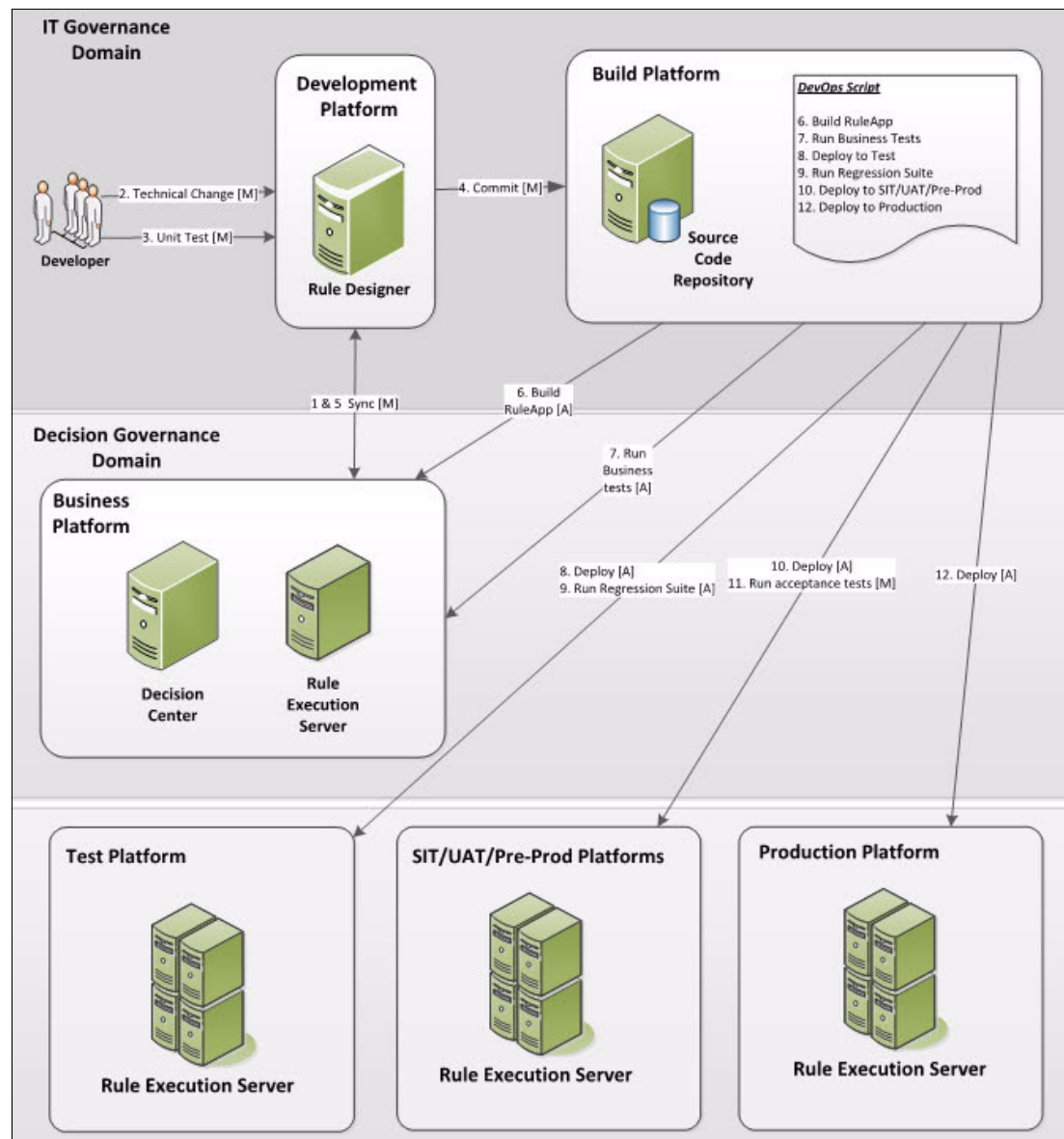


Figure 9-3 Business Centric DevOps: IT change

9.3 DevOps orchestration with UrbanCode Deploy

This section describes how to use the UrbanCode Deploy plug-ins to streamline your DevOps.

9.3.1 What is UrbanCode Deploy?

In a multi-software environment where ODM plays a small part in a big software stack, using a common tool to orchestrate the build and deployment is essential. This is where IBM UrbanCode Deploy (UCD) comes in.

UCD is a tool for automating application deployments through your environments. It is designed to facilitate rapid feedback and continuous delivery in agile development while providing the audit trails, versioning, and approvals needed in production.

For more information about UCD, see the [IBM UrbanCode Deploy overview](#).

9.3.2 The UrbanCode Deploy ODM plug-ins

The ODM Source Config plug-in for UCD can be used to build a RuleApp archive for a decision service rule project in Decision Center, and then to deploy the RuleApp archive to a target Rule Execution Server. If you want to build a RuleApp archive to be deployed at a later point in time, you can use the ODM Source Config plug-in to build the RuleApp archive and store it in the UCD internal repository called Code Station.

Then, use the Deploy Rule Archive step provided by the ODM plug-in for UCD to deploy your RuleApp archive to a target Rule Execution Server of your choosing. This second approach enables a RuleApp to be built once but deployed to many target Rule Execution Servers.

The ODM Source Config plug-in for UCD is documented in the [ODM source config tutorial](#).

The ODM plug-in for UCD is documented in the [ODM plug-ins tutorial](#).

9.4 Suggested APIs for DevOps

This section acts as a technical reference for implementing ODM DevOps using the patterns discussed previously. APIs suitable for an IT Centric DevOps are indicated with an [I], and APIs suitable for a Business Centric DevOps are indicated with a [B].

- ▶ Building and Deploying decision services in eclipse [I]
Rule Designer provides a tool that [automates the build and deployment](#) of decision service rule projects.
- ▶ Synchronizing from Rule Designer to Decision Center [I or B]
You can use the synchronize Ant task to [automate the publishing and updating](#) of projects from Rule Designer to Decision Center.
- ▶ Testing decision services [B]
[Test decision services](#) against tests suites held in Decision Center.

- ▶ Deploying decision services from Decision Center [B]
[Deploy decision services](#) from Decision Center using the Decision Center API.
- ▶ Deploying decision services with REST APIs [I or B]
In a cloud environment, the REST APIs are the preferred way to [Deploy decision services](#).
- ▶ Detecting a release is approved in Decision Center [B]
To detect that a release has been approved, create a custom session controller to override the `onCommitElement`. [Verify that the release has transitioned](#) from `Ready for Approval` to `Complete`:

9.5 Conclusion

In this chapter, we described two approaches to using DevOps. IT Centric and Business Centric. The Business Centric DevOps approach provides the greatest agility for most organizations.

Finally, we gave an overview of UrbanCode Deploy and provided references to ODM APIs that will help build your DevOps solution.



ODM on Cloud

IBM Operational Decision Manager (ODM) on Cloud is IBM's software as a service (SaaS) offering that provides decision service development and governance capabilities in the cloud platform.

IBM ODM on Cloud offers a subset of the capabilities of the full ODM product, but for many organizations these capabilities enable them to start using the technology quickly and is sufficient for their needs. It includes the following functionality:

- ▶ Rule Designer to build your projects
- ▶ A Decision Center environment for developing and deploying decisions
- ▶ Three environments for deployment:
 - Development
 - Test
 - Production

It enables your developers to create the project structures for your decision services, and your business users to create and test rules. In IBM ODM on Cloud, you can use either the decision governance framework or the manual branching capabilities of the product.

This chapter covers the following topics:

- ▶ Considerations
- ▶ IBM ODM components
- ▶ User roles
- ▶ Cloud environments
- ▶ Workflow
- ▶ IBM ODM on Cloud in a hybrid cloud environment
- ▶ Conclusion

10.1 Considerations

From a Governance point of view, there are a few considerations that users should be aware of:

- ▶ Only decision services are supported (no support for Classic Rule projects).
- ▶ Classic Rule Engine is not available, so there is no Decision Validation Service (DVS).
- ▶ No Java API or Ant scripts are available. Only the Rule Execution Server REST API is available for DevOps deployment operations.

There are other considerations, which can be found in the documentation but which have little (or nothing) to do with Governance as covered in this book.

10.2 IBM ODM components

Developing a project for the cloud involves the following three components:

Rule Designer	<p>A rich client tool that you download from the cloud development environment. Rule Designer is the main tool that is used by a rule developer to build the model and technical artifacts that are required for authoring and running a decision service.</p> <p>Rule Designer includes a local capability for running rules and setting up a solution before it is published to the development environment in the cloud. From Rule Designer, you can deploy to the development environment on the cloud, but not to the test and production environments. Rule Designer is the first tool in the lifecycle of a decision service.</p>
Decision Center	<p>A collaborative web tool and repository in the cloud, Decision Center is part of the development environment. It is used in the authoring and governance of decision services by a team that is led by a release manager. Decision Center includes the Business Console for all regular authoring, testing, governance, and deployment activities, and the Enterprise Console for occasional administrative and advanced activities.</p>
Rule Execution Server	<p>The runtime component that runs decision services and makes them available as web services to external applications, processes, or services. Each environment has its own isolated instance of Rule Execution Server. During its initial development, a decision service can be deployed from Rule Designer to the Rule Execution Server in the development environment.</p> <p>However, the decision service cannot be deployed from Rule Designer to the Rule Execution Server in the test environment or in the production environment. Instead, Decision Center provides a governance framework and strong tracing capabilities to control the deployment of decision services in the different environments: First in development, then in test, and finally in production.</p>

10.3 User roles

IBM ODM on Cloud has a slightly simplified set of user roles, shown in Table 10-1 and Table 10-2 on page 138.

Table 10-1 Business roles in IBM ODM on Cloud governance

Governance role	Responsibilities	Also known as
Release Manager	<p>Works primarily in the Business Console component:</p> <ul style="list-style-type: none"> ► Orchestrates the lifecycle of a decision service, and is ultimately responsible for the deployment of a decision service release to production. ► Follows a staged progression from development to test and production: <ul style="list-style-type: none"> – Creates development branches or releases – Defines change and validation activities for rule developers, business users, and integrators – Assigns ownership of work, reviews, and approvals ► With author rights, the person in this role can create deployment configurations in the Business Console for the development, test, and production environments. 	
Business User	<p>Works primarily in the Business Console component:</p> <ul style="list-style-type: none"> ► Implements and maintains some or all of the business rule artifacts that are in a decision service. ► Runs functional tests and simulations in the development environment to validate the changes that are made for a release. ► Can deploy a decision service to the test environment to validate changes in a test application. Can also deploy to the development environment. ► Can participate in the review or approval process for other business-user or integrator activities. 	<ul style="list-style-type: none"> ► Business Policy Architect ► Rule Analyst ► Rule Architect ► Business Policy Analyst ► Rule Author ► Rule Writer ► Business Rule Author ► Rule Validator ► Business Analyst
Integrator	<p>Uses the Business Console component in addition to other development, integration, and test tools:</p> <ul style="list-style-type: none"> ► Builds the applications that call a decision service in the development, test, and production environments. ► Can be involved in the validation activities that are defined by the release manager. ► Can deploy decision services to the development and test environments. ► Can view and use the decision services that are deployed in the production environment. 	
Permission manager	<p>Works primarily in Business Console:</p> <ul style="list-style-type: none"> ► Implements the security policy on decision services. ► Creates groups, sets the permissions, adds users to the groups, and sets the groups on decision services. ► Can create deployment configurations in the Business Console for the development, test, and production environments. ► Can deploy decision services in all environments. ► The permission manager can take on all the ODM roles, for example, when the team is small. 	

Table 10-2 Technical Roles in ODM governance

Governance role	Responsibilities	Also known as
Cloud Administrator	<p>This role manages users and assigns roles in the IBM ODM on Cloud portal:</p> <ul style="list-style-type: none"> ▶ There must be at least one administrator in an IBM ODM on Cloud subscription. ▶ Only another administrator can remove an administrator. ▶ Most users do not have a cloud administration role. 	
Rule Developer	<p>This role performs the following functions:</p> <ul style="list-style-type: none"> ▶ Designs the structure of the implementation to support the design requirements and evolution of the decision services. ▶ Develops technical functionality to support authoring (XOM and utility functions). ▶ Formalizes the business terminology (vocabulary) used in Operational Decision Manager. ▶ Makes the initial version of the business rule artifacts, including textual rules, decision tables, and ruleflows. ▶ Runs the decision service locally or in the cloud development environment until adequate results are achieved. ▶ Publishes the decision service from Rule Designer to Decision Center. ▶ Collaborates with the release manager and business users in authoring and governance activities. ▶ Collaborates with the integrator to integrate the decision service into an application. ▶ Creates deployment configurations in Rule Designer for the execution environments. 	

10.4 Cloud environments

IBM ODM on Cloud provides the following environments on the cloud to support a staged progression from development to production.

Access to these environments is controlled, and depends on the role of the user and the criticality of the environment.

Development environment

Users:

- ▶ Rule developers (in Rule Designer for ODM on Cloud)
- ▶ Release managers
- ▶ Business users
- ▶ Integrators
- ▶ Permission managers

Includes:

- ▶ Decision Center
- ▶ Development Rule Execution Server
- ▶ Rule Designer for ODM on Cloud

Used for:

- ▶ Collaborating on the development and lifecycle of a decision service
- ▶ Running decision services in development activities

- ▶ Performing functional tests and simulations that are run from Decision Center
- ▶ Testing a development application that calls a decision service that is still in development

Note: Although the Decision Center instance is in a Development environment, the service level agreement for ODM on Cloud makes it the right location for the purposes of architecture. FixPacks, and so on, are handled by the ODM on Cloud Software as a Service and you do not need a *Test* instance for Decision Center in this case.

Test environment

Users:

- ▶ Business users
- ▶ Integrators

Includes:

- ▶ Test Rule Execution Server

Used for:

- ▶ Testing a decision service during or after its development, and within the governance framework that is defined by the release manager:
 - Performance tests
 - System tests
 - Integration tests
 - Other user acceptance tests

Production environment

Users:

- ▶ Release managers

Includes:

- ▶ Production Rule Execution Server

Used for:

- ▶ Running the decision services in Production

10.5 Workflow

Figure 10-1 illustrates how the user roles in IBM ODM on Cloud interact with the product components during the lifecycle of a business rules application:

1. Rule development
2. Validation
3. Testing
4. Promotion of the release in the cloud production environment

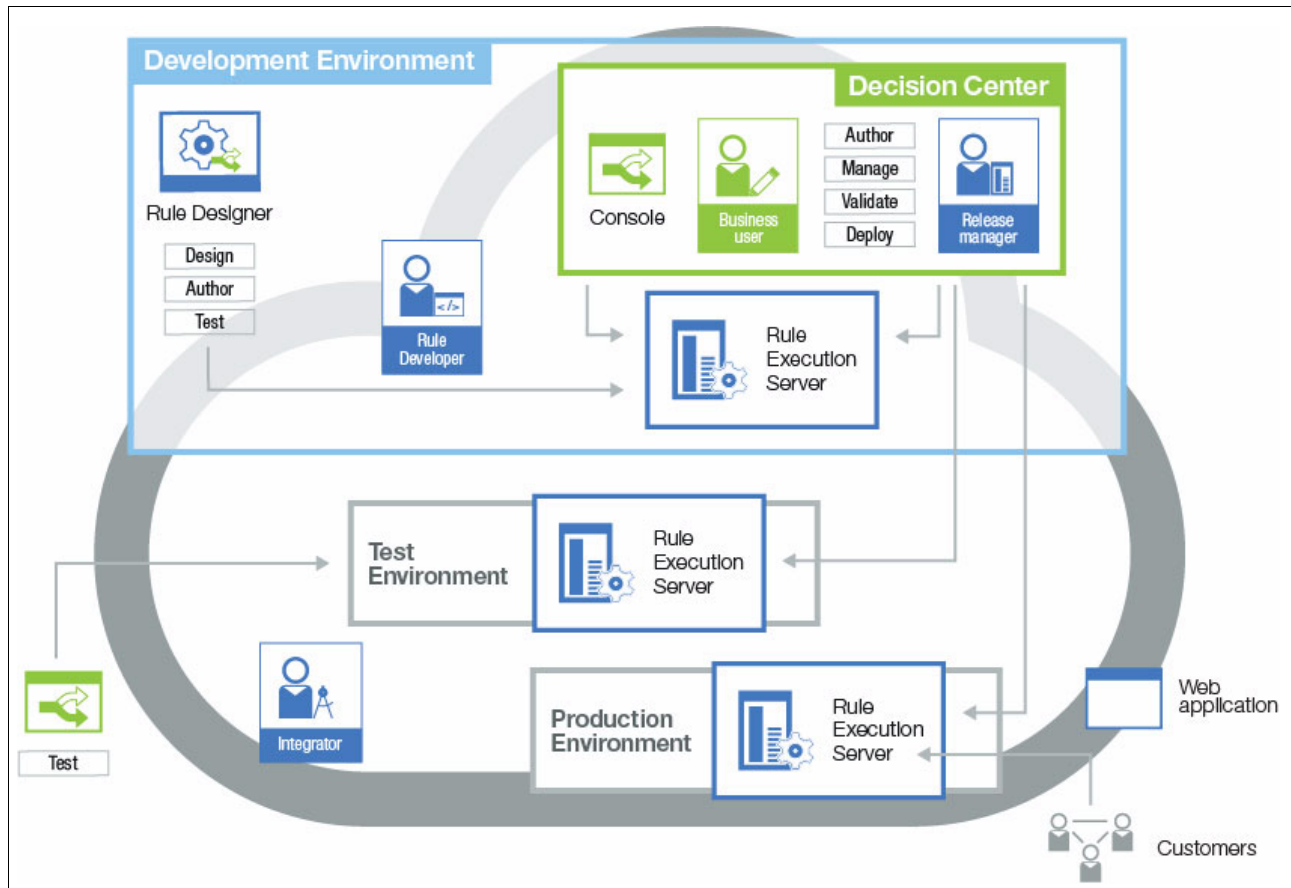


Figure 10-1 ODM on Cloud Workflow Diagram

10.5.1 Setting up the user roles

The first person to access the IBM ODM on Cloud product is the Cloud Administrator. This person might not use ODM, but must invite the appropriate people and assign them roles.

10.5.2 Collaborating to develop the decision service

The rule developer uses Rule Designer to write and edit rules, deploy the decision service to the development environment, and publish the decision service in Decision Center. This person is competent in Java and understands the company object model and works with the software architects to develop the first version of the decision service. They define the vocabulary and write the business rules that model business logic. The rule developer publishes the initial version of the decision service to Decision Center.

The release manager then creates an additional release with change activities by using the decision governance framework in the Decision Center console. They set up change activities and validation activities in Decision Center. When the team has completed the validation activities, and the rules are ready to test in the development environment, they deploy the completed release to the Rule Execution Server.

The business user views and edits the rules through the Business Console. The work is tracked as a change activity that records the modifications that are made to the rules. The business users can also perform some tests using the testing framework and the Excel spreadsheets.

The permission manager enables security and user access settings in the console to make sure that the business user, and other users who need access to the release and activity branches in the decision service, are assigned to the correct user groups.

10.5.3 Testing and promoting the decision service

The release manager has access to all of the cloud environments so that they can perform testing throughout the development lifecycle of the decision service. For example, when the team is satisfied with the behavior of the decision service in the development environment, they start the next step in the workflow by deploying the most recent decision service release to the test environment.

The integrator tests and validates that the decision service is running as expected by working in the Business Console and in all of the cloud environments. They monitor and evaluate the function of the decision service when it is called from a client application. To run the tests, they add some code to the application so that it calls the new pricing decision service that was developed in IBM ODM on Cloud. The client application runs on an external application server and calls the decision service running in the IBM Cloud test environment.

The release manager evaluates the performance of the decision service in the test environment before deploying it to production. He works with the Integrator to run benchmarks using standard software development validation procedures.

When the development is complete and the Release Manager is satisfied with the benchmark results, they promote the decision service to the Rule Execution Server in the production environment. The release manager is the only user role that has access rights to deploy a decision service to production.

10.6 IBM ODM on Cloud in a hybrid cloud environment

The concept of hybrid cloud designates the connection of one or more clouds to on-premises systems or to other clouds.

A hybrid cloud environment combines technologies that run simultaneously and complement each other on traditional corporate on-premises systems and on the cloud. Hybrid cloud provides flexible allocation of systems while maintaining high levels of security, availability, and performance.

IBM ODM on Cloud supports hybrid cloud environments by allowing deployment to Operational Decision Manager Rule Execution Servers with compatible versions that are hosted outside of the IBM ODM on Cloud environment. For example, deployment can take place on an enterprise on-premises system, in a private cloud, or on IBM Bluemix®.

10.7 Conclusion

The IBM ODM on Cloud provides an environment that organizations can use to quickly get started with the ODM technology. It provides most of the functionalities you would have with an on-premises installation of ODM, and has a built-in Governance Framework that prescribes a workflow for your decision governance. In this chapter, we described some of the main elements that are important to know for ODM on Cloud.

Limitations:

- ▶ Only decision services are supported (no support for Classic Rule Projects).
- ▶ Classic Rule Engine is not available, so no Decision Validation Service (DVS).
- ▶ No Java API or Ant scripts, which might limit some DevOps operations.

ODM on Cloud uses a simplified set of pre-defined roles and includes a new one for Cloud Administration.

The execution environments are preset to:

- ▶ Development
- ▶ Test
- ▶ Production

It is possible to use ODM on Cloud in a hybrid cloud environment.



Branching and merging

Although the focus of this IBM Redbooks publication is to work with the decision governance framework, readers should be aware that it is possible to have decision governance by using branching and merging. This chapter provides some details about how this can be achieved.

Note: It is important to note that one of the major benefits of using the decision governance framework is that it enforces a workflow that is designed to minimize issues with development, testing, and deployment. By using branching and merging, this workflow is not enforced in any way.

This chapter covers the following topics:

- ▶ Branching and merging to manage releases
- ▶ Deployment events
- ▶ Branching strategies
- ▶ Example of branching and merging in IBM Operational Decision Manager (ODM)
- ▶ When to use branching and merging
- ▶ Going from branching and merging to Decision Governance Framework
- ▶ Conclusion

11.1 Branching and merging to manage releases

This section introduces concepts that might be familiar to technical users but not to business users. Branching and merging are concepts that are borrowed from source code control in software development.

11.1.1 Branching and merging

The underlying principle for branching and merging is that you must have traceability of what specific changes were performed. We want to answer the following questions:

- ▶ Which rules were changed?
- ▶ Which decision table values were changed?
- ▶ What were the changes?
- ▶ Who performed them?

In traditional software development, the answers to these questions are found in the source code control system (SCCS) used by the development team. Decisions are no different in that you must be able to have the answers to all those questions. In IBM ODM, that functionality is provided by the repository and is where branching shows its value.

Branching

To understand branching, you must understand the concept in more depth.

First, you must consider the structure of the decision services in packages and rules, which are like folders and files on a computer. The packages (folders) provide a structure for the rules (files). Creating a branch is like taking a copy of the packages and rules and putting that copy in a different location so that it does not interfere with the original copy. In this case, the original copy is called the *main branch* or *trunk*.

If you were to make a copy with directories and files on your computer, you could then make changes to any file within the copy without interfering with the original version of the files.

As you make these changes to the copy (the branch), you start enhancing some of the original documents and improving them. These changes are not part of the main branch, but it usually makes sense to have the main branch benefit from these enhancements. This is where merging comes in.

Merging

The goal of merging is to have the changes from a branch merged into the main branch so that the main branch can benefit from these changes. Merging changes is a complex task:

- ▶ Rules that were not changed in the branch do not need to be merged.
- ▶ Rules that did not exist in the main branch must be created.
- ▶ Rules that are deleted in the branch might need to be deleted in the main branch (you must be careful about this task).
- ▶ Rules that were changed in both the main branch (trunk) and the branch can be in conflict, which must be resolved.

Although it is a tricky task to perform, it is important that a merge is performed appropriately so that the rules remain consistent and behave according to expectations. After a merge is complete, there is no need for the branch anymore, because the change history is shown in the history of the rules of the main branch.

11.1.2 Snapshots

Snapshots capture the state of a branch at a specific moment in time. Figure 11-1 shows the snapshot icon in the Business Console interface.

You can create snapshots of the different kinds of branches. You can consult and compare snapshots and, if you have the appropriate permissions, rename or delete existing snapshots. When you consult a snapshot, you cannot edit its contents.

You can restore a snapshot so that it becomes the current state of a branch. To restore a snapshot of a release, you must be an administrator.

Deployment snapshots are a special type of snapshot, which can be automatically taken at the moment of deployment, that captures the state of the rules at the moment of deployment. Deployment snapshots appear in the list of snapshots, and can be used to redeploy but these snapshots cannot be restored to make it the current state of a branch.

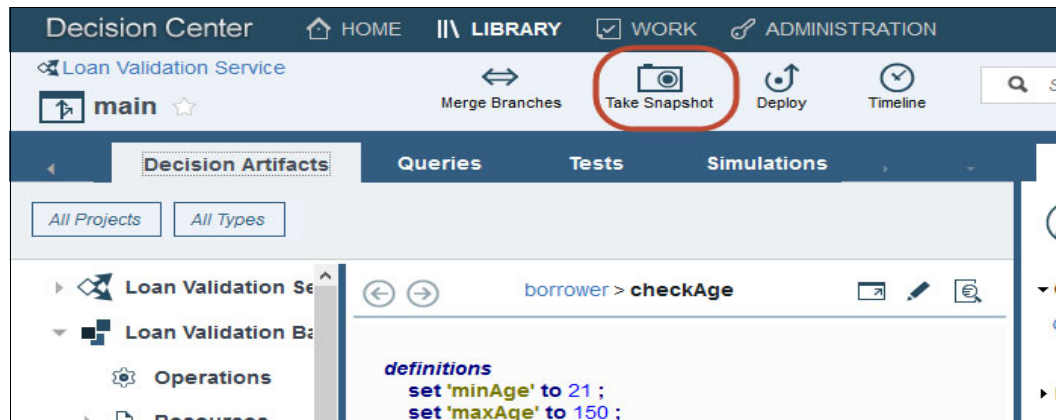


Figure 11-1 The snapshot button in Business Console

11.1.3 Naming conventions

To make it easier to track branches and snapshots over time, it is important that you establish a naming convention for them so that you can easily identify the appropriate one you might be looking for.

For example, snapshots can simply be named *Snapshot* followed by the current date and time: Snapshot 2017-05-05 15h50.

Similarly, a branch could follow a naming convention by week: Week of 2017-05-05.

You also want to consider the names for regular changes, emergency changes, long running changes, and so forth.

11.2 Deployment events

In this section, we provide details about two main scenarios that concern the deployment events and the resulting versions of runnable services so that we can describe the branching and merging strategies' ability to support these scenarios.

11.2.1 Scenario A: Single executable service

From the execution perspective, if your organization needs to only have a single execution service, then Scenario A might describe your requirements.

From an execution perspective, there is only one version of the decision service evolving at a time. The following events exist for this approach:

- ▶ Version 1 Deployed: This is the first executable version of the decision service and which is available for the client application to call.
- ▶ Version 1 Emergency Fix Deployed: If an issue were discovered in the rules in version 1.0, an emergency fix might be required and would be deployed.
- ▶ Version 1 Update Deployed: As time goes by, a normal update to the rules is deployed for execution.

Figure 11-2 shows the events and the available services over time.

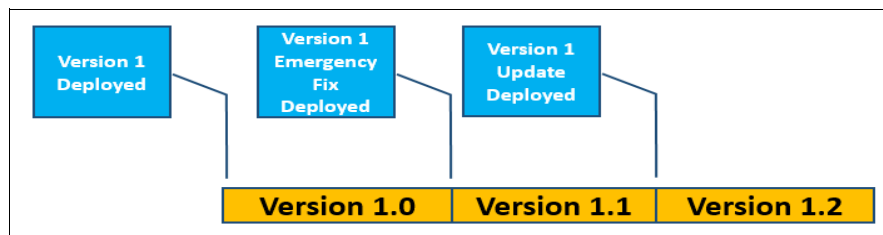


Figure 11-2 Scenario A: Single executable service

From Scenario A, branching and merging needs to support the following scenarios:

- ▶ Day-to-day development
- ▶ Emergency fix development on a specific deployed version

11.2.2 Scenario B: Parallel executable services

From the execution perspective, if your organization needs to have two (or more) execution services running in parallel, then Scenario B might describe your requirements.

From an execution perspective, there are multiple versions of the decision service evolving at a time. The events described for Scenario A apply, but we also need to be able to support additional events. The following additional events are specific to this approach:

- ▶ Version 2 Deployed: This is the first executable version of the parallel decision service and which is available for the client application to call.
- ▶ Version 2 Update Deployed: As time goes by, a normal update to the rules is deployed for execution on the parallel service.

Figure 11-3 shows the events and the available services over time.

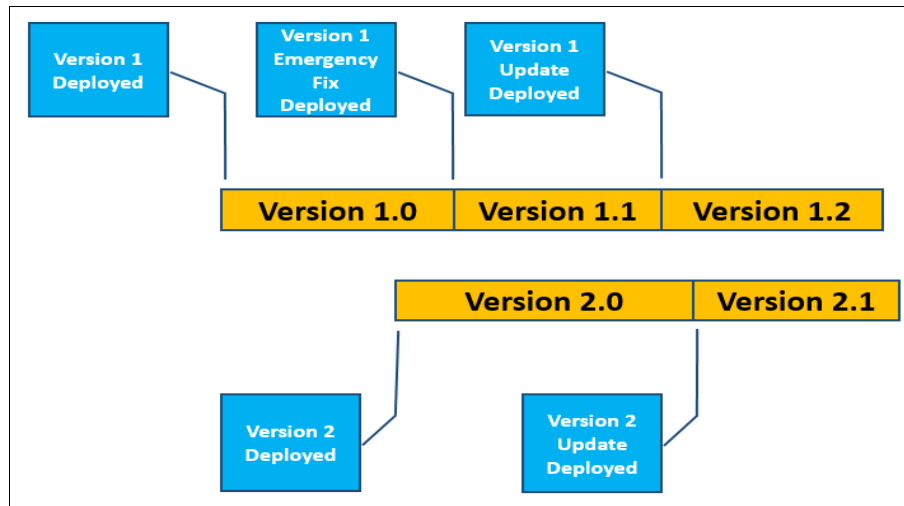


Figure 11-3 Scenario B: Parallel executable services

From Scenario B, branching and merging needs to support the following scenarios:

- Day-to-day development on Version 1
- Emergency fix development on a specific deployed version
- Day-to-day development on Version 2

11.3 Branching strategies

The following section provides details about some possible branching strategies that you can consider for your organization, or start from and adapt to fit your specific requirements.

For each strategy, you want to consider how well it supports the requirements from the two scenarios described in the previous section.

For each strategy, we look at the following information:

- How day-to-day simple changes are performed and managed
- How fixes on the currently deployed production version can be handled
- How major changes are performed and managed
- How well it supports development and fixes on multiple parallel versions deployed for execution

Although testing is an important part of the work required, in order to keep the test lighter, it is assumed that branches are tested successfully before being merged, and that the resulting branch after the merge is also tested successfully before any deployment takes place.

Note: In the diagrams that follow, the orange line shows the deployed version of the release in an execution environment. The labels shown in these diagrams are for demonstration purposes only, and do not reflect actual version numbers seen for the deployed ruleapp or ruleset.

11.3.1 Main branch with snapshots only

The most basic way to use the branching capabilities is to not use branches at all, and to only use snapshots to take pictures of the state of the rules in the main branch at a point in time.

It is assumed that after the first synchronization, a snapshot is taken, and that at each deployment, a snapshot is taken.

Scenario A analysis

This method of working can easily be used for day-to-day development on Version 1 as shown in Figure 11-4:

- ▶ Snapshots are taken regularly, especially before a deployment.
- ▶ There is a single branch, so no merging is required.
- ▶ The current state of the branch is deployed.

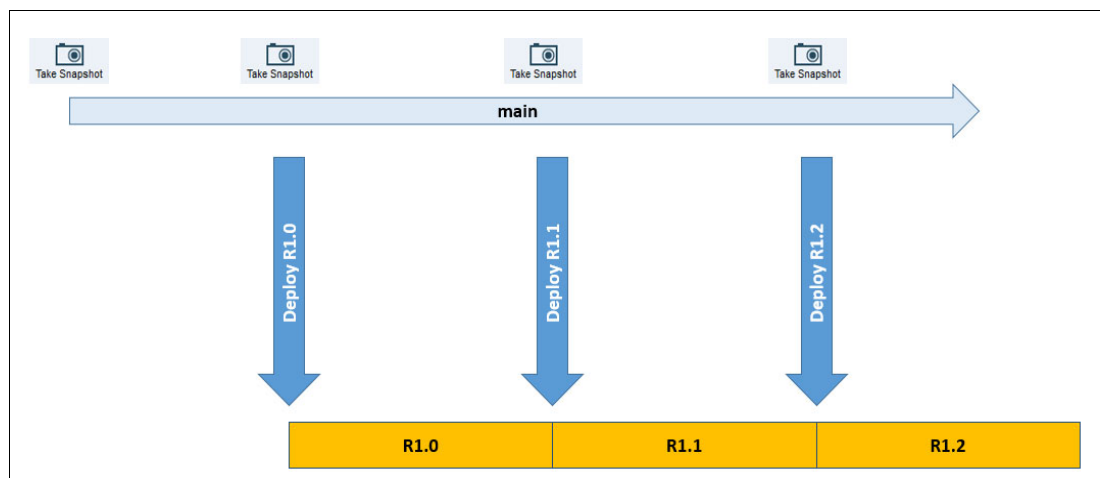


Figure 11-4 Main branch with snapshots only

Figure 11-5 on page 149 shows that an emergency fix requires more work:

1. Take a snapshot of the current branch state (undeployed).
2. Restore the snapshot of the latest deployment (the deployment that requires the emergency fix).
3. Make the changes to the rules and test that the change is working.
4. Deploy the Emergency fix (and take a snapshot of that deployment).
5. Restore the snapshot of the “current state (undeployed)”.
6. If applicable, reapply the changes to the rules so that the fix implemented in the emergency fix is also applied to the current state.

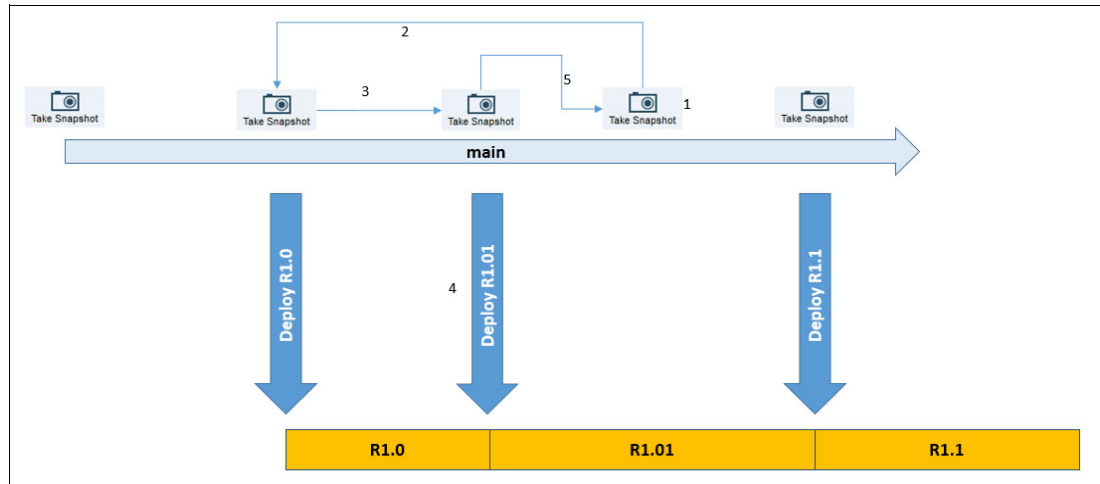


Figure 11-5 Main branch with snapshots only: Emergency fix

With this approach, major changes are handled the same way as day-to-day changes. This makes it a challenge to handle changes that take a long time because it would prevent deployment of simple day-to-day changes.

Scenario B analysis

The fact that branches are not used makes it impossible to work this way with a Scenario B situation.

Pros

Simple to implement.

Cons

The cons:

- ▶ Emergency fixes need to be done twice, duplicating that portion of the work (merging is not available because the approach uses a single branch).
- ▶ Relies heavily on snapshots, which is a manual process and increases complexity.
- ▶ Makes the implementation of major changes that take a long time difficult.
- ▶ Does not support Scenario B.

Although it is simple to implement, we do not recommend that you use this approach in a production environment. It was included in this book for the purposes of demonstration.

11.3.2 Main branch with simple branches for changes

The next level of branching is to use one main branch and to create sub branches from this branch. This approach allows day-to-day changes to be made in a branch that is not deployed. The main branch always represents the state of the rules as deployed in production.

Scenario A analysis

This method of working can easily be used for day-to-day development on Version 1. The day-to-day changes are made on a branch that is independent.

When completed, the changes can be merged into the main branch and then deployed, as shown in Figure 11-6.

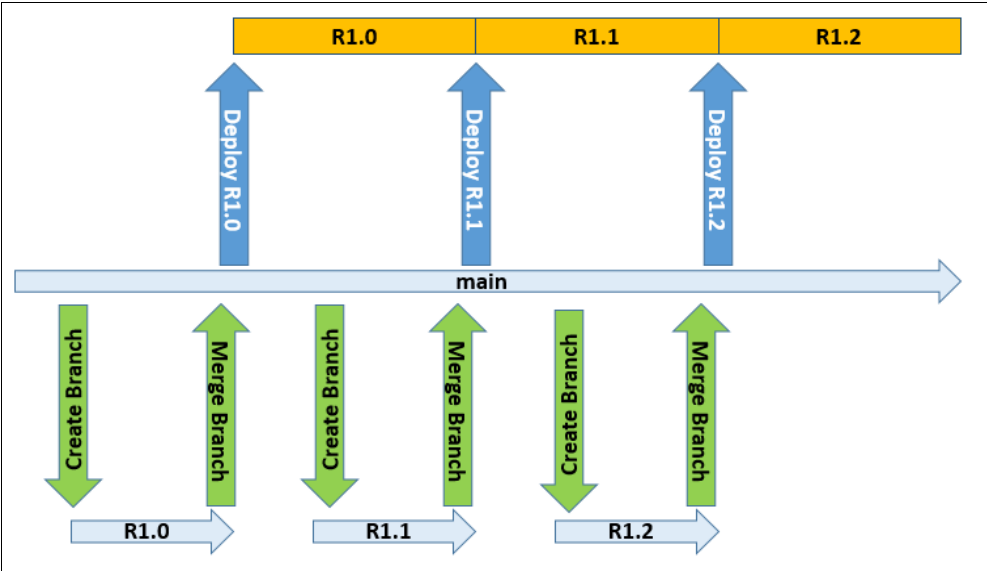


Figure 11-6 Main branch with simple branches for changes

Figure 11-7 shows how an emergency fix would be handled. The change can be performed directly on the main branch, which holds the current state of the rules in production. Alternatively, an emergency fix branch can be created from the main branch to allow the work to be performed there before being merged in the main branch and then deployed to production. The emergency fix also needs to be merged into the day-to-day branch.

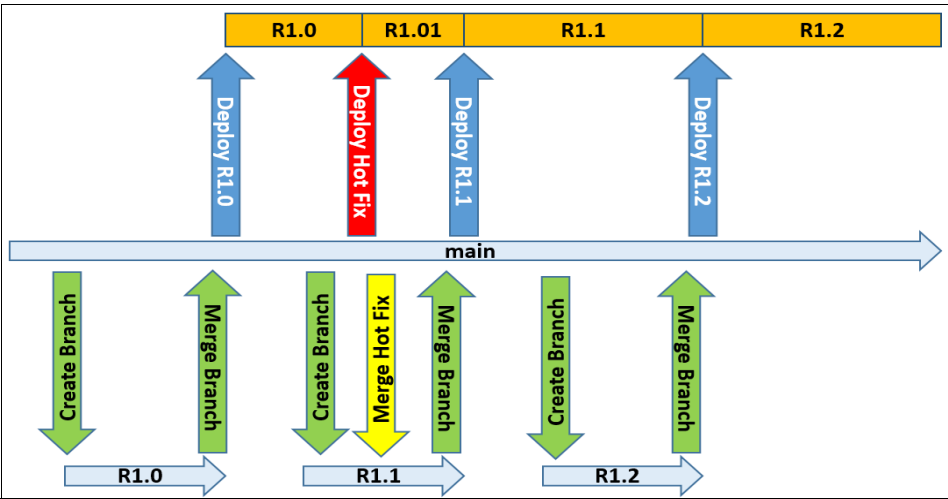


Figure 11-7 Main branch with simple branches for changes and an emergency fix

Major changes are handled the same way as day-to-day changes. A branch is created for the major change. If the change is very lengthy, day-to-day branches must be merged into the major change branch to keep it as current as possible.

Leading practice: As all changes are merged back into the main branch, there is no real need to keep any branches other than the main branch. When the merge has been completed, you can delete the “working branch”.

Scenario B analysis

Because there is only one branch that represents the state of deployed rules, it is impossible to work with this approach with a Scenario B situation.

Pros

The pros:

- ▶ Simple to implement
- ▶ Emergency and day-to-day changes easily handled
- ▶ No longer relies on snapshots
- ▶ Long-running changes are possible

Cons

Does not support Scenario B.

11.3.3 One branch per decision service version

This strategy is an extension of the “Main branch with simple branches for changes” strategy because it works the same way, but with one branch per decision service version.

Scenario A analysis

This strategy works the same way as the strategy from 11.3.2, “Main branch with simple branches for changes” on page 149, as shown in Figure 11-8.

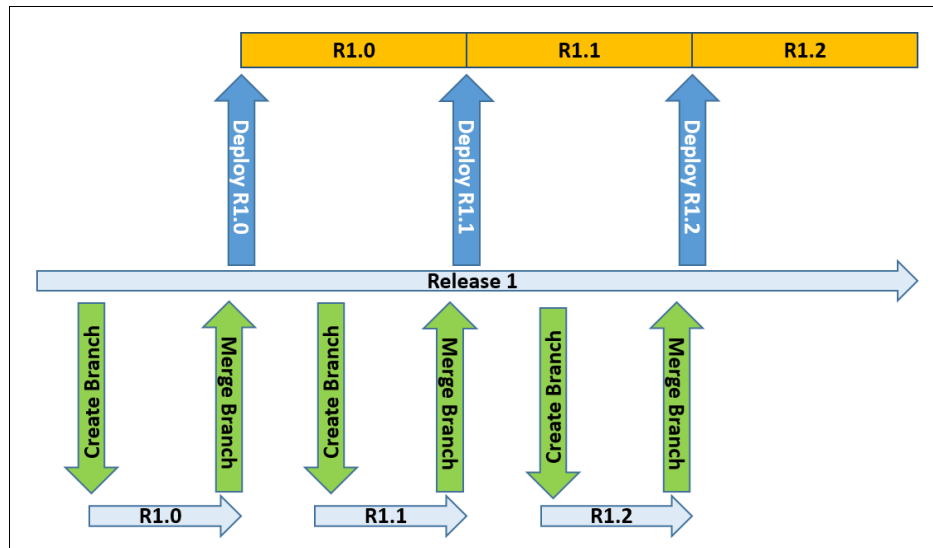


Figure 11-8 One branch per decision service version

What is not shown in Figure 11-8 is that there is a “main” branch from which the Release 1 branch was started.

Scenario B analysis

Because there is one branch per version of the decision service, this strategy also supports Scenario B in the same manner as Scenario A.

Pros

The pros:

- ▶ Medium complexity to implement
- ▶ Emergency and day-to-day changes easily handled
- ▶ Does not rely on snapshots
- ▶ Long running changes are possible
- ▶ Supports Scenario B

Cons

No major con identified.

This strategy is the suggested strategy for branching and merging in Decision Center.

11.3.4 Decision Governance Framework like branching

If you want to explore branching and merging a bit further, you can try to mimic what the decision governance framework does with branching and merging, as shown in Figure 11-9. First, you want one branch per release of the decision service, enabling you to support Scenario B. Each Release can have one or more change activities, and each Change Activity is represented as a sub-branch of the release.

When a Change Activity is completed, it is merged back into the release branch. When all of the Change Activities for a release have been merged into the release branch, the release can be deployed.

To keep Figure 11-9 simple, we did not show the “main” branch or the major release branches. Major release branches enable multiple versions of the decision service to be worked on and deployed in parallel, and support Scenario B. When a release is completed, the changes are merged back into the major version branch.

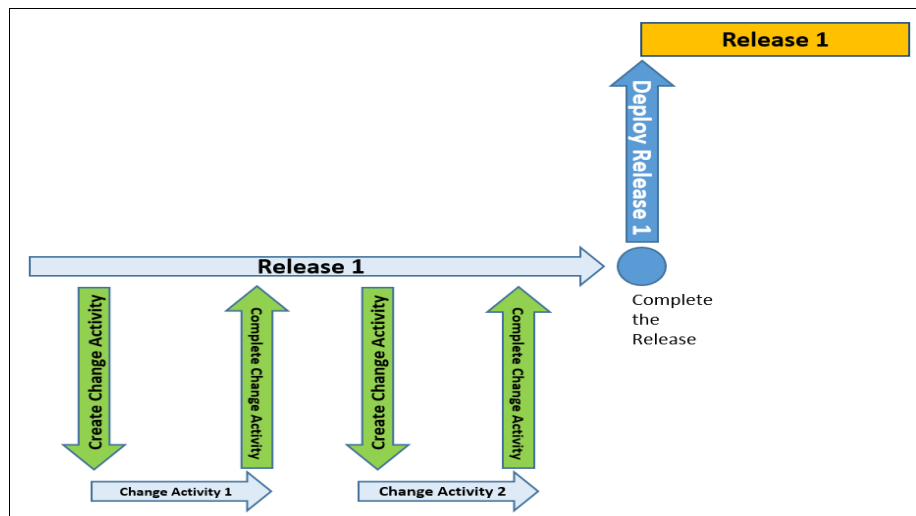


Figure 11-9 Decision governance framework Branching and Merging

This strategy is similar to the strategy described in 11.3.3, “One branch per decision service version” on page 151. The main difference seen here is that multiple change activities can be completed before the deployment of the release takes place.

Scenario A analysis

Scenario A is supported. The main difference is the use of multiple change activities before completing a release. Emergency fixes are supported in a similar way as the approach detailed in 11.3.2, “Main branch with simple branches for changes” on page 149.

Scenario B analysis

Because there is one branch per version of the decision service, this strategy also supports Scenario B in the same manner as Scenario A.

Pros

The pros:

- ▶ Medium complexity to implement
- ▶ Emergency and day-to-day changes are easily handled
- ▶ Does not rely on snapshots
- ▶ Long-running changes are possible
- ▶ Supports Scenario B

Cons

No major con identified.

11.3.5 Advanced branching strategies

It is possible to implement some even more advanced branching and merging strategies, but these are not discussed in this book. If you decide to implement a more complex strategy, it is important to test its functionality and the workflow that you need to have so that it is a viable long-term solution.

11.4 Example of branching and merging in IBM Operational Decision Manager

This section shows an example of the “One branch per decision service version” branching and merging strategy. All of the operations are shown in the Business Console component of IBM Operational Decision Manager. The example also assumes that the user is logged in to the application with the appropriate rights to perform the operations that are indicated.

The starting point is immediately after the first synchronization to Decision Center from Rule Designer. Figure 11-10 shows the user interface where you can see that the main branch was created. The Plus sign (+) at the top of the list of branches enables you to add new branches.

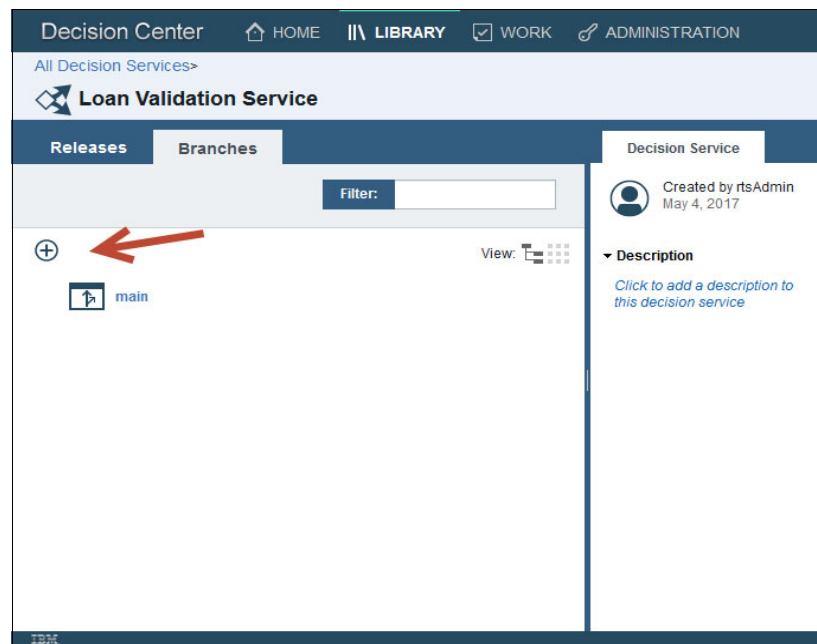


Figure 11-10 After the initial synchronization

To implement this strategy in IBM ODM, complete the following steps:

1. The first step is to create the branch for the main version of a release. In this example and as shown in Figure 11-11, we call it Release 1 and base it on the main branch.

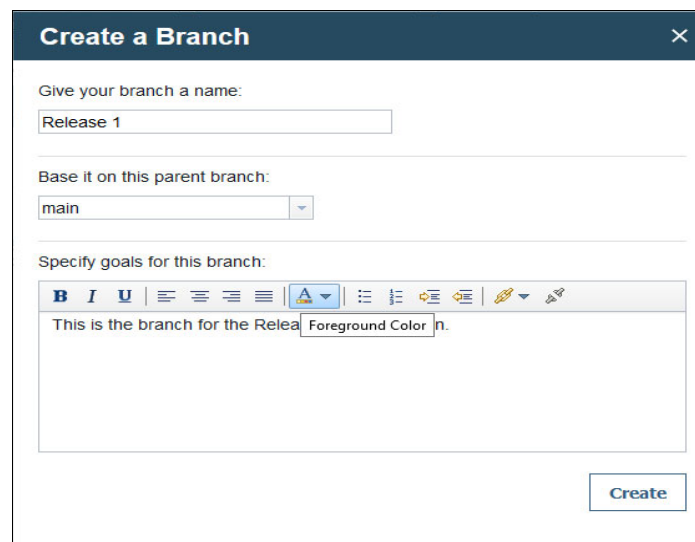


Figure 11-11 Creating a major Branch

If you need to work on another major release at the same time, repeat these steps for the other major release. This process is not described in detail in this book. Figure 11-12 shows the result of the creation of this first major release.

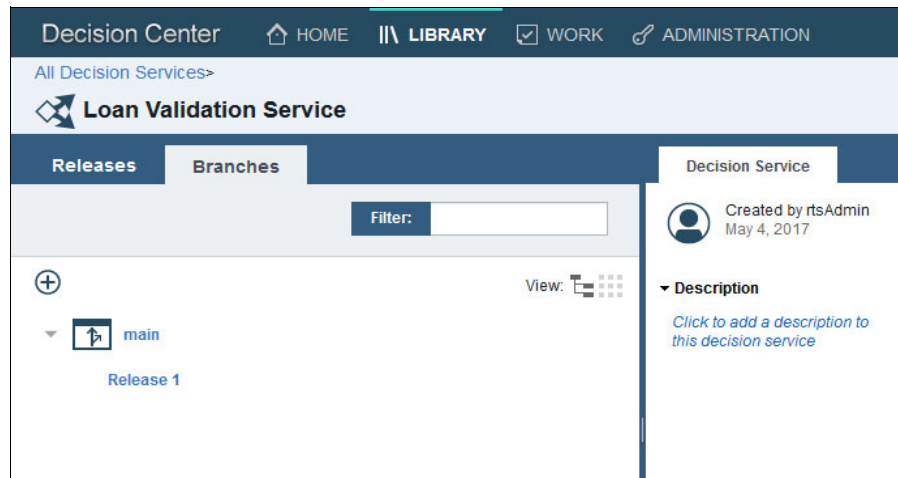


Figure 11-12 Release 1 is created

2. Next, create a minor release sub-branch based on this major release branch so that changes can be managed in that sub-branch before being merged back. The result is shown in Figure 11-13.

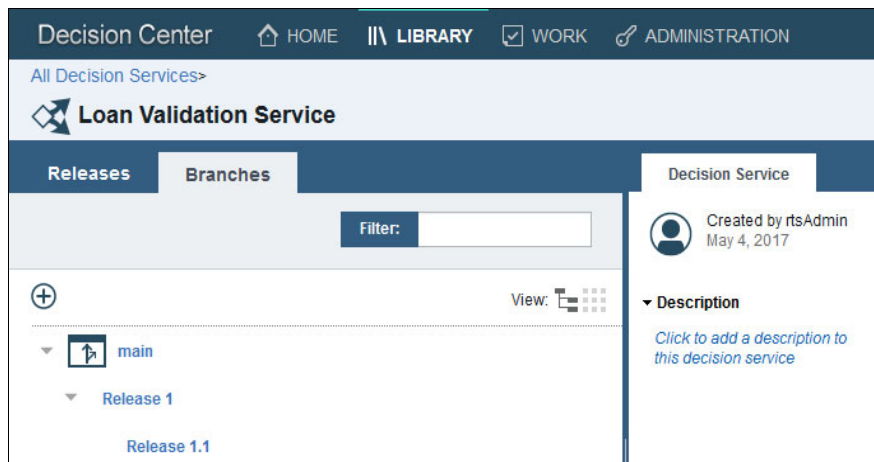


Figure 11-13 Release 1.1 is created

At this point, we expect that multiple changes in the rules for the decision service might take place over time. We show one simple change as an example.

Figure 11-14 on page 156 shows the interface where a rule is selected for edition. First, notice the upper left corner of the user interface, which confirms that you are modifying the Loan Validation Service, and that you are currently working in the sub-branch called Release 1.1. On the left side of the screen you can see the list of projects for the decision service.

The borrower package is currently selected, showing the rules that it contains on the middle portion of the screen (Figure 11-14).

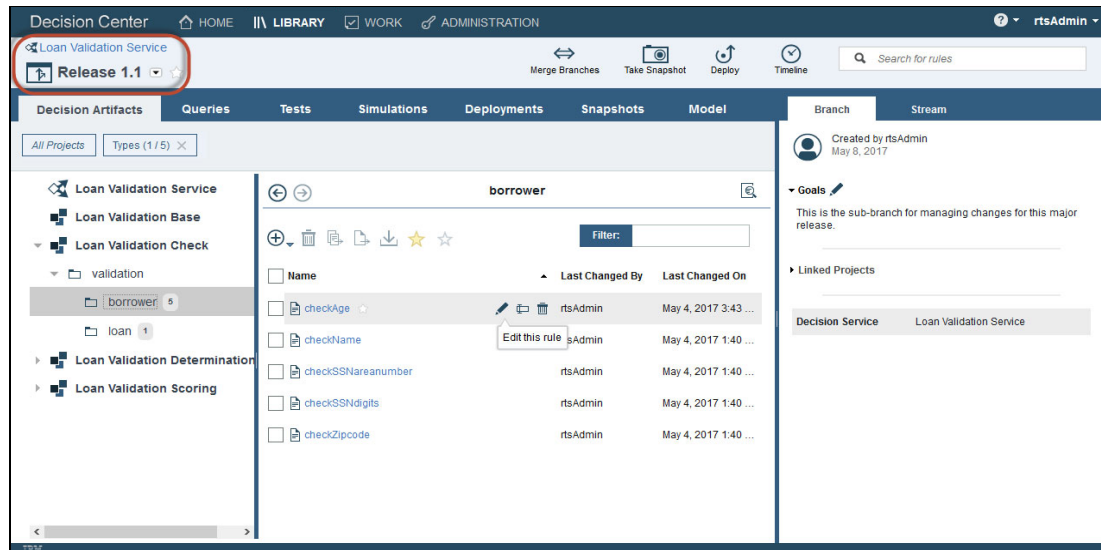


Figure 11-14 Selecting the rule to edit

3. You can edit the rules, in this case making sure that the ages accepted for the rules are correct. When you save your change, you are prompted for a comment to add to the new version that you just created, as shown in Figure 11-15.

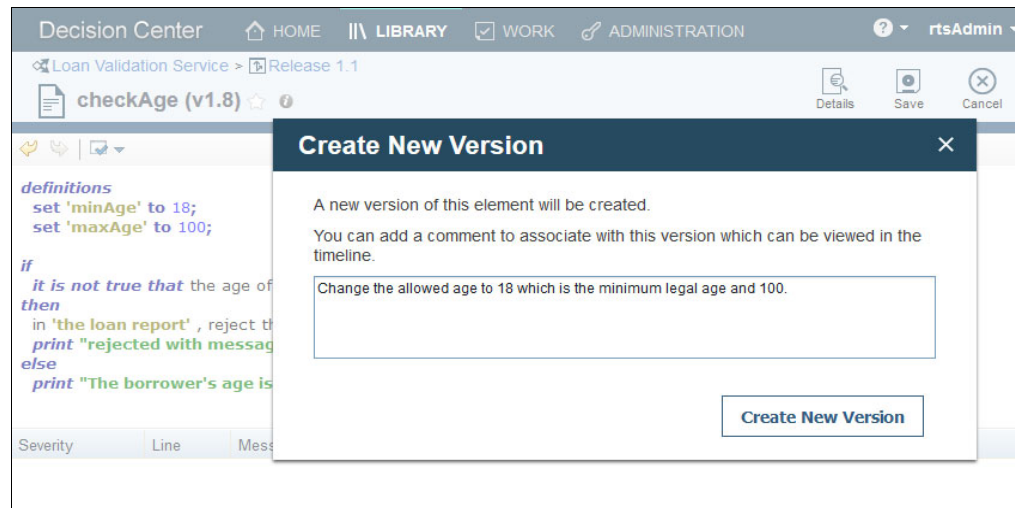


Figure 11-15 Creating a new version of a rule

More changes can be performed in the rest of the rules. When all of the changes required for this minor release are completed, it is time to look at the sub-branch of Release 1.1 and to merge it back to the major branch, as shown in Figure 11-16.

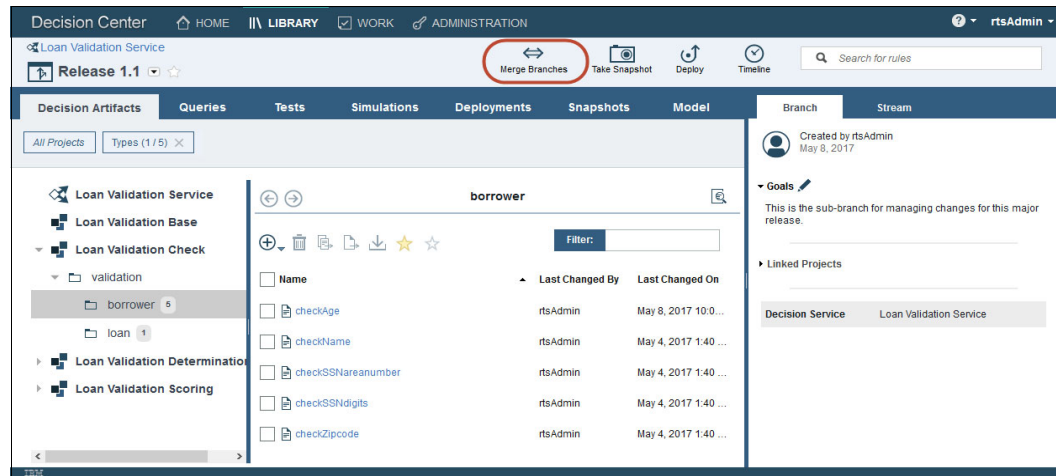


Figure 11-16 Ready to merge changes from the sub-branch into the major branch

4. This opens a screen that allows you to choose which branch you want to merge your changes to. Select the Release 1 major branch and then click **Merge**, as shown in Figure 11-17. As a good practice, all changes in either branch should be stopped during this time, and the option to lock the branches to perform the merge should be selected.

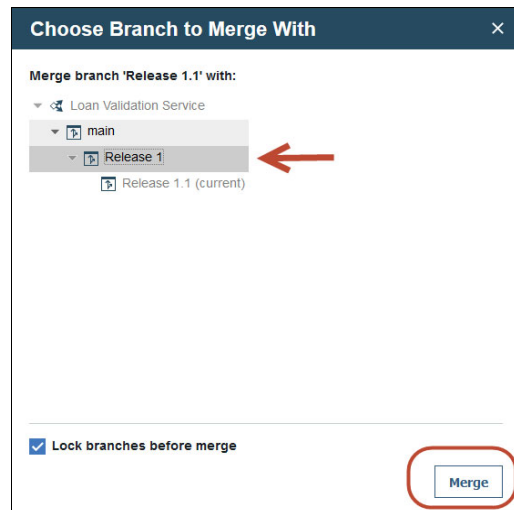


Figure 11-17 Choosing the branch to merge with

- The resulting screen shown in Figure 11-18 shows that two differences were identified. You are prompted to perform an operation, which is to **update in Release 1** in both cases in this example.

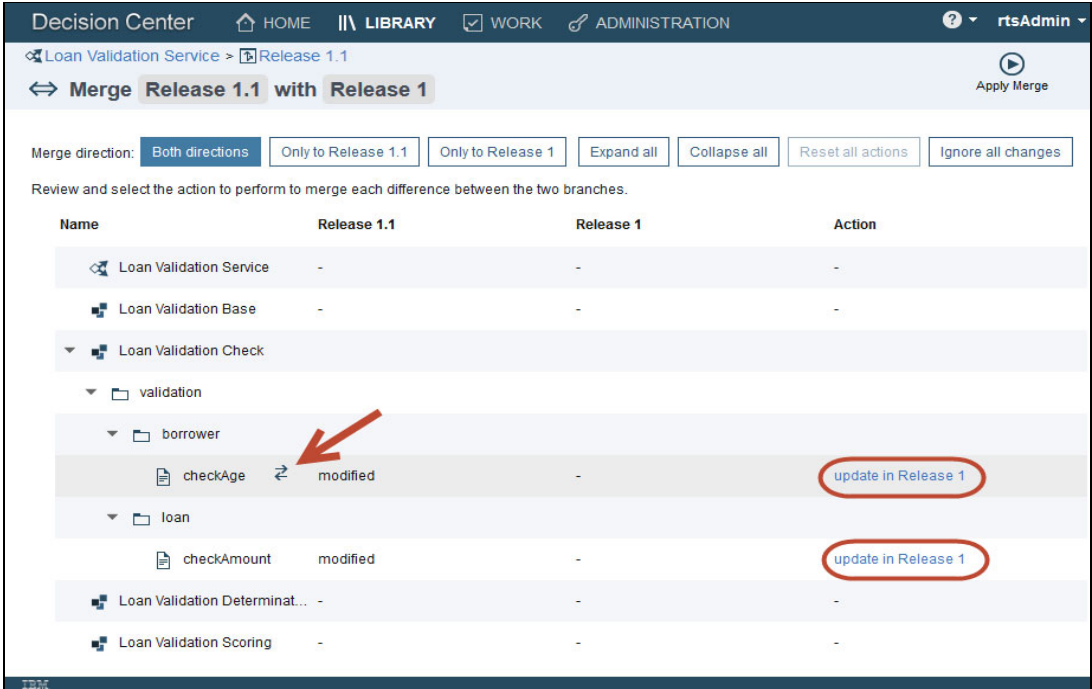


Figure 11-18 Differences between branches

- As you hover your mouse over one of the changes, it is highlighted and a small icon displays that enables you to compare the rule from both releases before performing the merge. The comparison screen is shown in Figure 11-19.

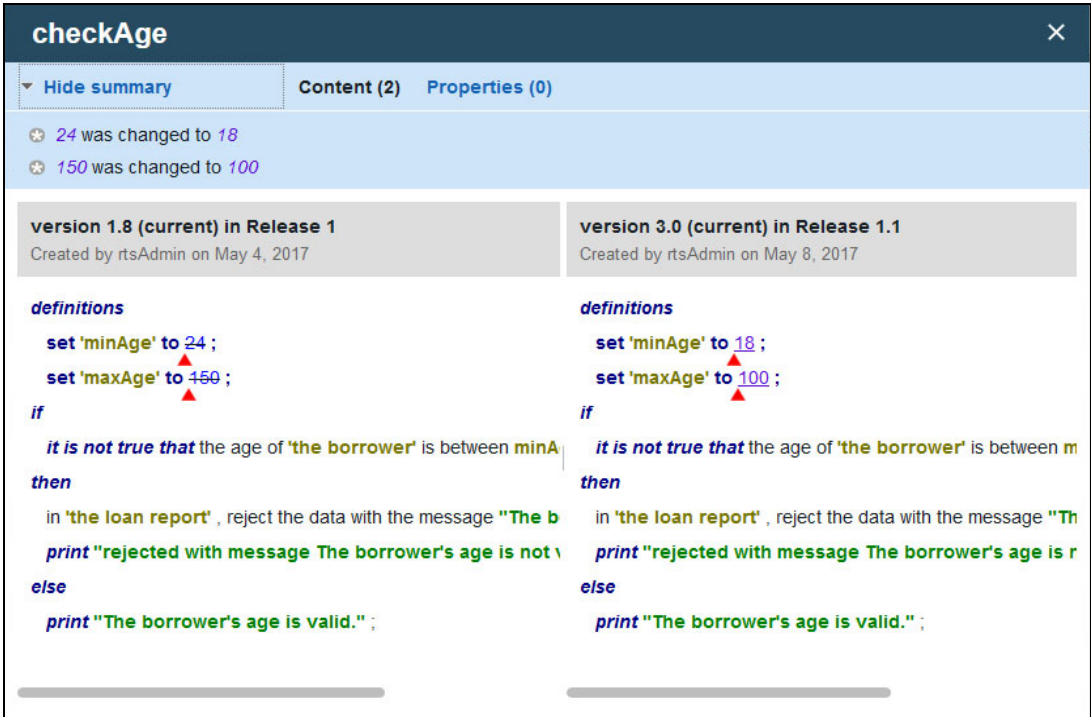


Figure 11-19 Comparing the changes in two versions of a rule

At this point, review all the changes to ensure that they are as required. You have the option to go back and fix some of the changes, change the way the merge will be performed, and then apply the merge using the icon at the upper right of the screen.

7. In some cases, it might happen that the same rule was modified in the two branches being merged, which results in a conflict that is displayed in the merge screen, as shown in Figure 11-20. Compare the two rules and then decide on the option to take, which in this case is **do not modify**. Alternatively, you can replace the version in either the major branch or the minor branch, depending on which one needs to take precedence. Determining which version is implemented is a business decision.

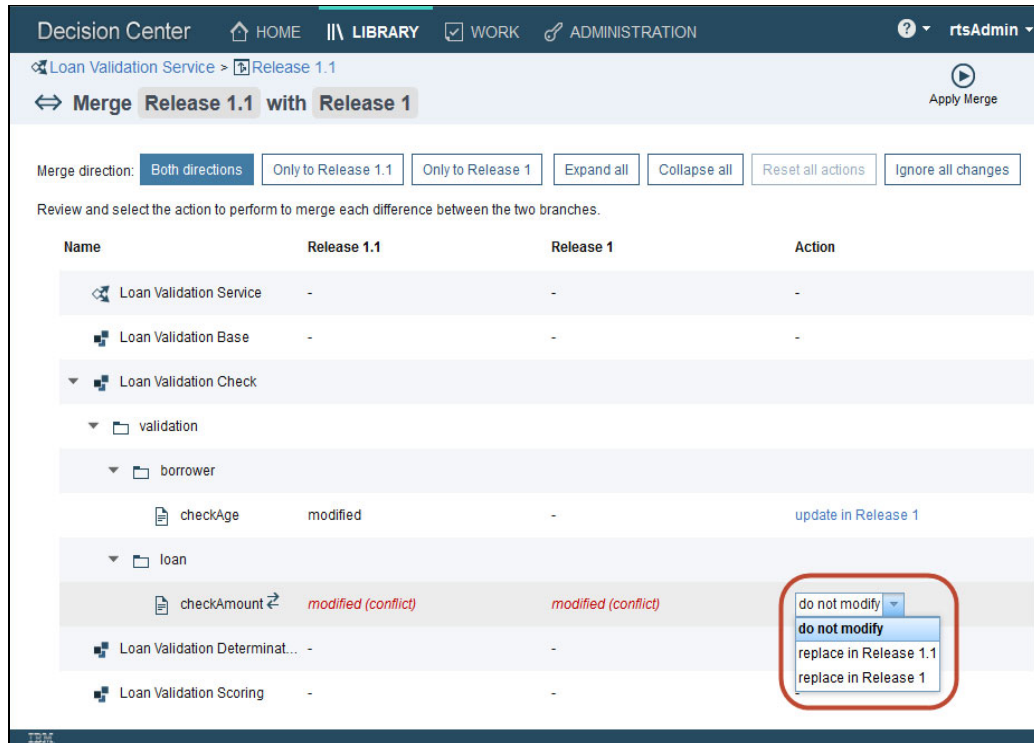


Figure 11-20 Conflict detected during merge

8. When you are satisfied with the choices selected for the merge, apply the merge and complete details, as shown in Figure 11-21.

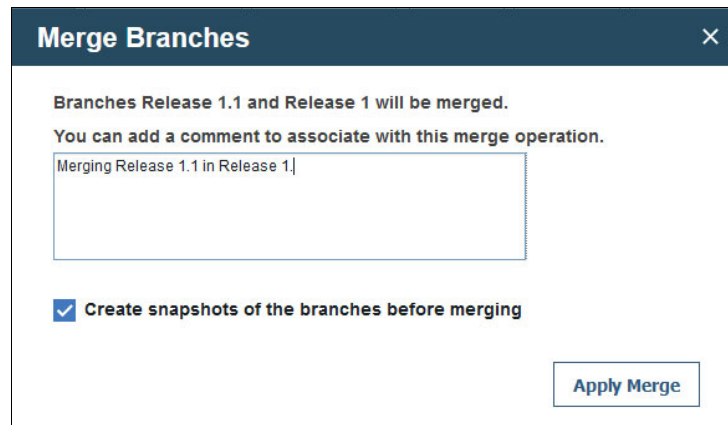


Figure 11-21 Applying the merge

Note: In this case, we suggest that you select the **Create snapshots of the branches before merging** option so that automatic snapshots of each branch are taken.

9. After the operation is completed, you receive confirmation that the operation was successful, as shown in Figure 11-22 .

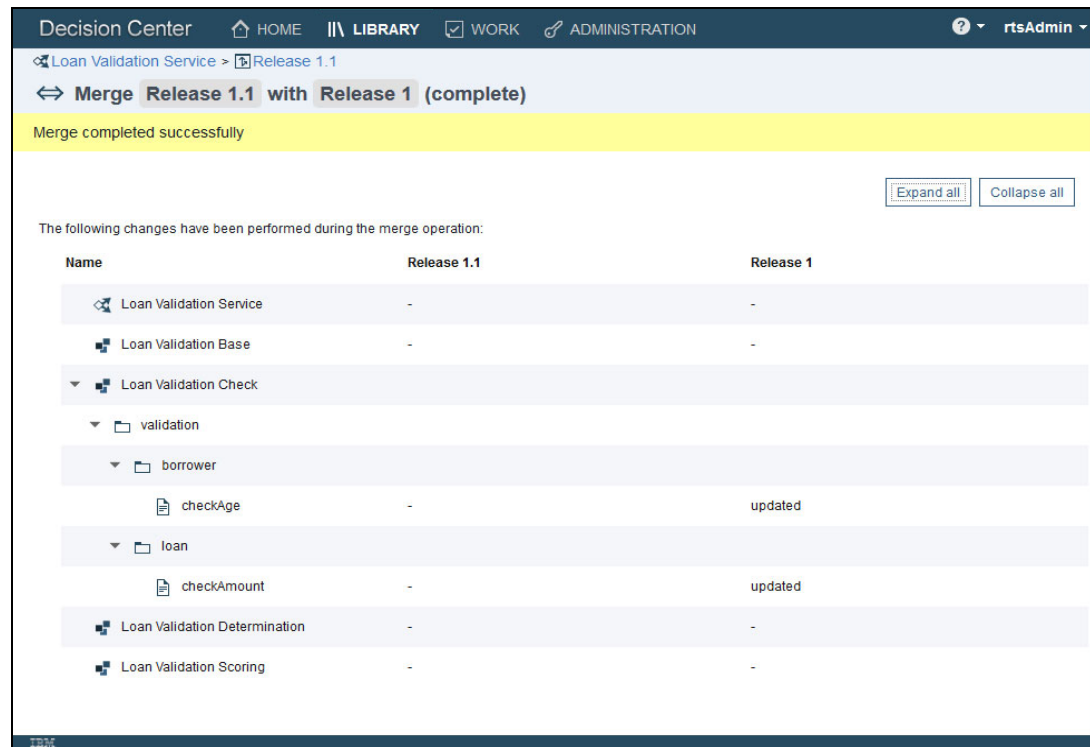


Figure 11-22 Merge completed successfully

At this point, a new sub-branch for the minor release "Release 1.2" can be created.

11.4.1 Leading practices

Here are some leading practices to consider:

- ▶ Create a separate branch for each major version (release) and one sub-branch per minor version (change activity).
- ▶ Try not to have parallel sub-branches unless they deal with separate sets of rule changes (modifying the same rule in multiple parallel branches increases the complexity of merging).
- ▶ Prepare your naming convention before implementation and “try it out” to make sure that it works for your organization.
- ▶ Think about the workflow and how you might manually do what is enforced by default when using the decision governance framework.

Develop a merging strategy:

- ▶ If you are using multiple sub-branches in parallel, regularly merge changes from the release branch into each sub-branch to determine potential conflicts early.

- ▶ Only when everything works in the sub-branch should you consider merging back to the release branch.
- ▶ Make sure that someone is playing the role of release manager, so someone manages the releases and the merging operations.

Leaving the main branch behind

The concept of releases that was introduced in 5.2, “Project hierarchies” on page 64 requires us to question the relevance of the main branch for decision management. Releases represent a unit that can be deployed as a decision service. It is possible to have multiple concurrent releases that focus on some specific decision changes that must be kept together:

- ▶ Consider each release its own *major branch*, which we will call the *release branch* (created from the main branch). With this approach, each release is independent from other releases, and multiple releases can be managed in parallel. If required, changes from one release can also be merged into another release.
- ▶ If the Release branch becomes the *major branch*, change activities then must be managed as part of sub-branches that are created from the release branch.
- ▶ If your organization has only a single release at a time, using the main branch might still make sense.

11.5 When to use branching and merging

The decision governance framework was created to provide a structured way to manage your decision services. Specifically, it manages your changes by using Releases, Change Activities, and Validation Activities. This framework makes users follow a simple workflow to reduce issues with the deployment of your decision services.

This simple workflow does not exist when using branching and merging. Permissions regarding who can work on rules as part of a change activity and so on are also not enforced.

Here are some factors that might affect your choice to use branching and merging as opposed to the decision governance framework:

- ▶ Number of rule authors
- ▶ Frequency of expected changes
- ▶ IT-centric or Business-centric processes
- ▶ Need for a simple workflow to be enforced
- ▶ Control of changes with permissions

Figure 11-23 shows a quick comparison of decision governance framework with the different branching strategies discussed in this chapter.

	Decision Governance Framework	Main branch with snapshots	Main branch with simple branches	One branch per Decision Service Version	"DGF-like" branching
Enforced governance workflow	X				
Control of changes with permissions	X				
Automatic merge of change activities	X				
Deployment to production controlled	X				
Manual merge of branches	X		X	X	X
Supports Scenario A - Single Service only	X	X	X	X	X
Supports Scenario B - Parallel Services	X			X	X
Long running changes possible	X			X	X
Complexity of Emergency Changes	Easy	Complex	Medium	Medium	Medium
Complexity of implementation	Easy	Medium	Easy	Medium	Medium
Relies on manual snapshots	No	Heavily	No	No	No

Figure 11-23 Comparison of decision governance framework to branching strategies

11.6 Going from branching and merging to Decision Governance Framework

After using branching and merging for a while, you might find yourself in a position where you want to take advantage of the decision governance framework:

1. Ensure that all the changes in your branches are merged into the branch that you want to use as a starting point.
2. In Rule Designer, create a new workspace.
3. Select **File** → **New** → **Project**.
4. Select **Rule Project From Decision Center**.
5. Enter the connection information and connect.
6. Select the decision service project and the branch that you want to connect to and synchronize.
7. Right-click the decision service and select **Decision Center** → **Disconnect**.
8. Select **Delete current connection entries**.
9. Make sure that all dependent projects (such as the XOM) and all components are in a valid state with no errors.
10. Right-click the decision service and select **Decision Center** → **Connect**.
11. Enter the connection information and connect.
12. Click **Next**.
13. On the next screen, select **Use Decision Governance Framework**.
14. You are prompted to confirm that you want to create the Initial Release required for the Decision Governance Framework. Click **Yes**.
15. Click **Finish**.

This creates the Initial Release in the decision governance framework, and you are now able to create the first “working” release using the framework.

11.7 Conclusion

This chapter describes how to use Branching and Merging should your organization decide not to use the decision governance framework. Here are some key takeaways from what was discussed:

- ▶ Create one branch per major release of your decision service.
- ▶ Create one sub-branch from the major release branch per minor release of your decision service.
- ▶ Branching and merging does not enforce a workflow as the decision governance framework does.



Conclusion

This IBM Redbooks publication focused on the governance of operational decisions within the enterprise. We explained how to configure IBM Operational Decision Manager (ODM) to enable safe yet agile change to business policies. We explained that the difference between IT Governance and decision governance is that decision governance puts the power of change into the hands of the business. This enables agility and ultimately a more agile, reactive, and competitive organization.

Because decision governance requires more collaboration between IT and business, in Chapter 2, “Decision governance for project managers” on page 9, we provided guidance to help organizations manage decision governance projects. Then in Chapter 3, “Roles and responsibilities in governing decisions” on page 21 we took a detailed look at the roles required for decision governance.

Chapter 4, “Securing the Decision Center” on page 31 explained how to create the roles and users described in Chapter 3, “Roles and responsibilities in governing decisions” on page 21. It also described how to assign the appropriate permissions to be able to perform their duties.

Chapter 5, “Designing decision services” on page 61 provided leading practice implementation suggestions so that the artifacts within a decision service can be managed in a scalable way.

Chapter 6, “Processes” on page 73 described the processes for business and IT change. It explained that IT-centric processes describe the governance of change by technical users and Business-centric processes describe the governance of change by business users.

Chapter 7, “Decision governance framework” on page 85 provided a detailed look at the built-in framework for Business-centric changes within ODM.

Chapter 8, “Deployment” on page 119 detailed the IT-centric and Business-centric approaches to deployment, and discussed the concept of two-phase deployment that can be used when Decision Center Rule Execution Server are disconnected.

Chapter 9, “ODM DevOps” on page 127 described how the content in Chapter 6, “Processes” on page 73, Chapter 7, “Decision governance framework” on page 85 and Chapter 8, “Deployment” on page 119 can be automated using ODM APIs.

Chapter 10, “ODM on Cloud” on page 135 looked under the covers of IBM’s cloud solution for ODM.

Finally, Chapter 11, “Branching and merging” on page 143 explained how to use the advanced repository management features of Decision Center when the decision governance framework is not used.

12.1 Where you go from here

This book provided leading practice governance suggestions and process templates that can be tailored for a governance solution that works in your company.

Once your governance solution is up and running, it is important to elicit feedback from users of the governance processes. Maturity and experience build over time, and capturing feedback will help improve the solution. Additionally, IBM is continuously releasing new versions of the product. It is important to make a cost/benefit analysis on the cost of upgrading versus the benefit of new features that will make your governance solution more streamlined.

In Chapter 2, “Decision governance for project managers” on page 9 we talked about setting up a Center of Excellence (CoE). The first users will be the ODM pioneers who will be the champions supporting the next project. After a few projects, the pioneers will develop a community to share ODM leading practices and patterns. The funding would initially come out of individual projects, but eventually there might be a specifically funded role to lead the CoE. The CoE would then become a group of SMEs enabling agile business policy change throughout the organization.



A

Acronyms

This appendix provides the meanings of and descriptions for the acronyms that are used in this book, as shown in Table A-1.

Table A-1 Acronyms

Acronym	Meaning	Description
ADM	Architecture Development Method	The ADM describes a method for developing an enterprise architecture, and forms the core of TOGAF.
BOM	Business Object Model	The BOM is the basis of the vocabulary that is used in business rules.
BPA	Business Policy Analyst	
CCB	Change Control Board	A committee that makes decisions regarding whether proposed changes to decisions should be implemented.
CoE	Center of Excellence	Center of Excellence can help an organization through a change and help the people build the skills and knowledge they need to fill their new roles.
COBIT	Control Objectives for Information and Related Technology	A framework that was created by ISACA to provide IT governance and management in an organization.
DVS	Decision Validation Service	Part of the IBM Operational Decision Manager offering that is used to test and simulate rulesets in Rule Designer and Decision Center.
ITIL	Information Technology Infrastructure Library	A set of practices for IT service management that focus on aligning IT services with the needs of a business.

Acronym	Meaning	Description
RES	Rule Execution Server	Part of the IBM Operational Decision Manager offering that provides a reliable and scalable execution environment for your business rule application.
SaaS	Software as a Service	A software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted.
SCCS	Source Code Control System	A system that allows you to track and control changes to source code documents.
SIT	System Integration Testing	Testing that focuses on a decision services coexistence with other systems
SLA	Service Level Agreement	A contractual agreement on the level of service to be provided by a service provider to a customer.
SME	Subject Matter Expert	Expert in a specific subject.
SOA	Service-oriented Architecture	An approach and a methodology to design software solutions that are based on interoperable services.
TOGAF	The Open Group Architecture Framework	A framework for enterprise information architecture (designing, planning, implementing, and governing).
UAT	User Acceptance Testing	Testing that is performed by a user to confirm that a decision service meets the requirements.
UCD	Urban Code Deploy	A tool for automating application deployments through your environments
XOM	Execution Object Model	The XOM is the model that is used to run the rules in IBM Operational Decision Manager.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *Making Better Decisions Using IBM WebSphere Operational Decision Management*, REDP-4836
- ▶ *Building an Application with Decisions, Processes, and Insight*, TIPS0940
- ▶ *Decision Automation on IBM System z with IBM WebSphere Operation Decision Management V7.5*, TIPS0960
- ▶ *Flexible Decision Automation for Your zEnterprise with Business Rules and Events*, SG24-8014
- ▶ *Implementing an Advanced Application Using Processes, Rules, Events, and Reports*, SG24-8065

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- ▶ Boyer, et al, *Agile Business Rules Development*, Springer, 2011, ISBN 3642190405
- ▶ *Business Decision Maturity Model (BDMM)*, found at:
<http://www.kpiusa.com/index.php/The-Decision-Model/business-decision-maturity-model-bdmm.html>
- ▶ *Business Rules Governance and Management*, found at:
<http://www.primatek.ca/blog/?dlname=Primatek-BusinessRulesGovernanceManagement-WhitePaper.pdf>
- ▶ *CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033)*, found at:
<http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>
- ▶ Office of Government Commerce (OGC), *The Official Introduction to the ITIL Service Lifecycle*, TSO, 2007, ISBN 0113310617
- ▶ Office of Government Commerce (OGC), *ITIL Service Transition*, TSO, 2011, ISBN 9780113313068

- ▶ von Halle, *Business Rules Applied: Building Better Systems Using the Business Rules Approach*, John Wiley & Sons, 2002, ISBN 9780471412939
- ▶ von Halle, et al, *The Decision Model: A Business Logic Framework Linking Business and Technology*, Taylor & Francis, 2009, ISBN 1420082817

Online resources

These web pages are also relevant as further information sources:

- ▶ ABRD website guidance on governance:
http://epf.eclipse.org/bp/abrd_1.5.1_20100928/process.abrd.base/deliveryprocesses/abrd_governance_BF9D60E1.html?nodeId=464d376e
- ▶ COBIT official web page of the ISACA website:
<http://www.isaca.org/Cobit/pages/default.aspx>
- ▶ Default behavior for customized session controller:
https://www.ibm.com/support/knowledgecenter/SSQP76_8.9.0/com.ibm.odm.dcenter.ref.dc/html/api/html/ilog/rules/teamserver/model/IlrSessionController.html
- ▶ Deploy the decision service from Decision Center using the Decision Center API:
https://www.ibm.com/support/knowledgecenter/en/SSQP76_8.9.0/com.ibm.odm.dserver.rules.deploying/topics/tsk_res_deploy_rlapp_ant.html
- ▶ IBM Operational Decision Manager at the IBM Operational Decision Manager home page:
<http://www-03.ibm.com/software/products/en/odm>
- ▶ IBM Knowledge Center for Operational Decision Manager Version 8.9.0:
https://www.ibm.com/support/knowledgecenter/SSQP76_8.9.0/welcome/kc_welcome_odmV.html
- ▶ IBM UrbanCode Deploy:
<https://www.ibm.com/us-en/marketplace/application-release-automation>
- ▶ Managing resources by using the REST API:
https://www.ibm.com/support/knowledgecenter/en/SSQP76_8.9.0/com.ibm.odm.dserver.rules.res.managing/topics/con_res_restapi_rsrcmng_intro.htm
- ▶ Operational Decision Manager Training at IBM:
<https://www.ibm.com/services/learning/ites.wss/zz/en?pageType=page&c=N417864C33318F08>
- ▶ ITIL at the Axelos website:
<https://www.axelos.com/best-practice-solutions/itil>
- ▶ SCRUM reference:
<https://www.scrum.org/>
- ▶ SOA Policy:
<http://www.ibm.com/developerworks/webservices/library/ws-soa-policy/index.html/>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



SG24-8395-00

ISBN 0738442658

Printed in U.S.A.

Get connected

