



68-BPR008

# Enrich the Decisioning of Your COBOL Applications

White Paper

# Enrich the Decisioning of Your COBOL Applications

White Paper

**Version 8.1**

Last Modified: June17, 2008

## Table of Contents

<b>1. Introduction</b> .....	<b>3</b>
Overview of Business Rules and Business Rule Management System .....	3
<b>2. Simple Steps to Business Rule Management for COBOL</b> .....	<b>4</b>
<b>3. Business Rule Management for COBOL Applications</b> .....	<b>7</b>
<b>4. Conclusion</b> .....	<b>8</b>

## 1. Introduction

Many articles and white papers have been written on why COBOL is far from “dead” and how many core business applications still run on it today. Few of these publications, however, focus on how to increase these applications’ business value. Organizations that rely on COBOL in their mainframes continue to look for better ways to maintain their COBOL applications and make them more responsive to business needs.

A tested way to improve these systems is to externalize the development and management of their business rules. As the “intellectual property of your company,” your business rules are based on your organization’s business policies and objectives. This white paper explains how our business rule management system (BRMS) can benefit your business, in addition to improve the efficiency of maintaining your COBOL applications.

Almost every major industry – from finance to manufacturing – has real-time systems that rely on COBOL and major organizations still want to maintain their core business applications in COBOL on mainframe computers. These corporations continue to see value in running their core business applications on large-capacity mainframes and these applications represent approximately 180 billion lines of code, with 5 billion lines added each year.

A large percentage of this COBOL code performs basic functions such as data transformation, data movement and report formatting. It takes many lines of COBOL code to do these basic functions, which today’s programming languages either do not have to do or can perform with substantially less code. But amidst all this code are key company assets in the form of business rules. These rules have been embedded in COBOL code for years. Often organizations do not even know which rules are truly running their business applications. Rules embedded deep within COBOL applications long ago, with no current documentation or expertise on them, drive systems that still run businesses today.

An organization’s success depends largely upon its ability to respond quickly to today’s complex, ever-changing markets and regulatory climate. For instance, being able to implement new pricing models or change or add products can be a key differentiator for ensuring business success. But many organizations cannot react quickly to business changes, especially when their business rules are embedded in COBOL applications. Their key business policies and regulatory compliance information is locked inside multiple software systems, expressed in programming language and generally inaccessible to the business managers responsible for implementing business policies and rules.

Businesses today need a BRMS solution that supports business changes and cycles far more effectively than traditional methods and takes direct advantage of business expertise. A true BRMS separates business logic from code and changes it without impacting the rest of the application. A single change can be assessed, implemented and tested in a very short timeframe.

## Overview of Business Rules and Business Rule Management System

To help explain how a BRMS can benefit your organization, here are a few definitions:

- **Business policy** – Every organization has people responsible for setting the policies by which the organization does business. A business policy is a statement of guidelines governing business decisions. An insurer may have an underwriting policy, for example, that says, “*An existing customer can be eligible for discounts for additional policies if he or she is the main insurance holder and has no disputed claims.*”
- **Business rules** – The specific statements that enforce a policy are business rules. Policies are translated into business rules, the detailed conditions and actions that unambiguously enforce the policy. The business rules expand upon the policy by stating in detail the circumstances under which the policy is applicable and the actions that enforce it. A policy can be translated into many business rules. In the insurance underwriting policy described above, for example, the rules need to define the terms of the policy (for example, “*existing customer*” and “*disputed claim*”). Regional regulations may require the rules to vary from region to region and “*the amount of the discount*” may change over time, as underwriting experiences more or less risk.

- **Business rule management system (BRMS)** – A BRMS provides the software to define and manage business rules through the business rule life cycle. The IBM® ILOG® JRules BRMS provides functionality for authoring, testing, analyzing and deploying business rules. Its repository provides full management functionality for business rules, including versioning, baseline management, security and metadata. ILOG JRules decreases development time and costs, for which customers have reported savings of approximately 50 percent.

With ILOG Rules for COBOL and the ILOG JRules BRMS, business rules buried in COBOL can be externalized and managed while still being executed in business applications on your mainframe. ILOG Rules for COBOL delivers several benefits, including:

- Ability to manage and document business decisions executed in mainframe applications
- Ability to generate native COBOL from rules within the ILOG JRules BRMS
- Authoring rules for COBOL in business terminology
- Ability to share business rules with Java™ and other COBOL applications

## 2. Simple Steps to Business Rule Management for COBOL

Your mainframe COBOL applications should readily accommodate the changes and needs of your business. When there is an application change request, the likelihood of that request requiring changes to business rules is very high. Change requests from business users are typically for business rules in order to update business policy or address an objective.

Let's take as an example an insurance company that we'll call All People Insurance. The insurer has a mix of mainframe and Java applications and its policy administration system is a mainframe COBOL application. All People needs to change its policy administration system because of a recently introduced New Jersey state regulation. This legislation requires business rules in COBOL code to be changed, which will take a COBOL developer a few weeks to analyze and make all the necessary changes, test them and deploy new code. Now, if All People had externalized these rules with ILOG Rules for COBOL to manage them in ILOG JRules, the company would have been able to quickly determine the set of business rules relevant to applying the new regulation, make the changes and regenerate the rules' COBOL code.

The ILOG JRules BRMS is the key element for making this COBOL application agile. ILOG Rules for COBOL is compatible with IBM Enterprise COBOL and has been extensively tested on IBM mainframes.

As these steps show, it's easy to move to rule management with ILOG Rules for COBOL and ILOG JRules.

### Step 1: Import the COBOL copybook that contains the fields you need to author rules

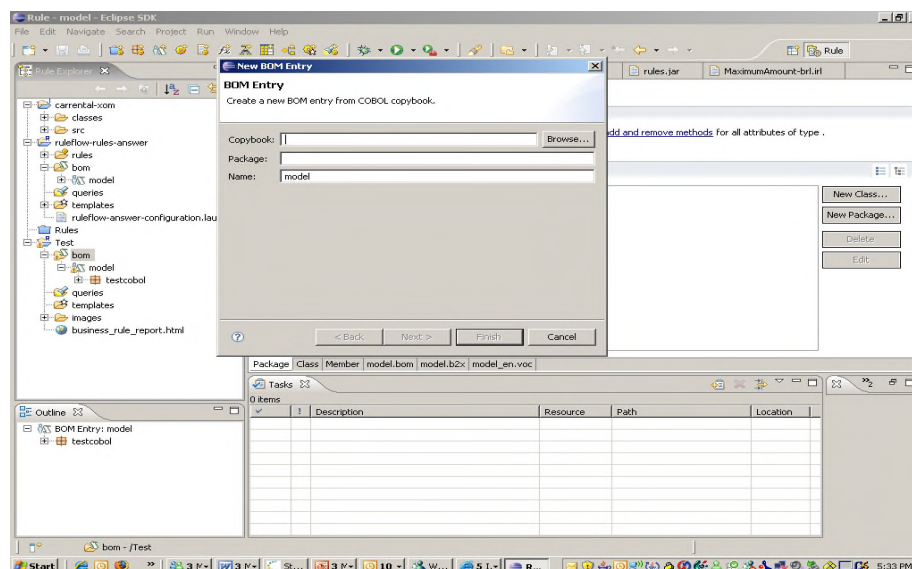


Fig. 1: ILOG Rules for COBOL interface for transferring rules from COBOL application

This step creates the business object model (BOM) needed to author the business rules. The source of the COBOL copybook identifies the mainframe fields in which the rules need to be written. Once the import is done, the business terms can be defined and used in the rule authoring instead of COBOL field names. For example:

COBOL Copybook with Mainframe Field Names	Business Terms For the Rules
P2002-CUSTFLD-A1	Preferred Customer Name
P2002-IDCTR	Previous Customer Status Indicator
P2003-CUSTFLD-A4	Customer Account Information

If you need to share these business rules with a Java application, you can create an execution object model (XOM). Otherwise, there is no need for one if you are using these rules to generate COBOL code.

## Step 2: Author the business rules

The screenshot displays the IBM ILOG Rule Studio interface. On the left, the 'Rule Explorer' shows a project structure with folders for 'rules', 'eligibility', 'validation', 'main', 'bon', 'queries', 'templates', and 'cobol'. Under 'rules', there are sub-folders for 'eligibility' (containing 'MinimumIncome' and 'MinimumCreditScore') and 'validation' (containing 'MaximumAmount'). A blue arrow points from the 'Risk by Debt to Income Ratio' rule in the explorer to a table in the main editor. The table, titled '\*Risk by Debt to Income Ratio', has columns for 'Loan Amount' (From, To), 'Yearly Income' (From, To), and 'Score points in risk factor'. It contains seven rows of data representing different risk levels based on loan and income thresholds. Below the table, a 'Business Rule: MinimumIncome' dialog is open, showing 'General Information' (Name: MinimumIncome) and 'Code' (a rule snippet: 'if the yearly repayment of 'the loan' is more than the yearly income of 'the borrower' \* 0.3 then add "Too big Debt-To-Income ratio" to the messages of the loan; reject the loan;').

Fig. 2: Organize and maintain rules in ILOG JRules

After the BOM has been created and the business terms have been provided, the business rules can be authored using IBM ILOG Rule Studio and IBM ILOG Rule Team Server:

- **ILOG Rule Team Server** is a highly scalable rule management server and repository with a collaborative Web environment for authoring, managing, validating and deploying business rules.
- **ILOG Rule Studio** is an integrated development environment (IDE) for rule applications that integrates directly into the Eclipse™ family of IDEs, including Eclipse and IBM Rational© Application Developer (RAD).

Each set of business rules is based on a business decision required for a business application. You can set up a rule project for each business decision that will in turn equate to one subprogram generated by ILOG Rules for COBOL.

You will need to design a rule flow for your ILOG Rules for COBOL project. The rule flow will be used to identify the order and dependencies in which the rule sets are needed for generation and execution in the COBOL subprogram.

Now that you have externalized and authored your business rules and saved them in the ILOG JRules rule repository, they are ready to be changed with your business needs. You can quickly respond to a business change by rapidly updating the applicable business rules.

### Step 3: Generate COBOL code

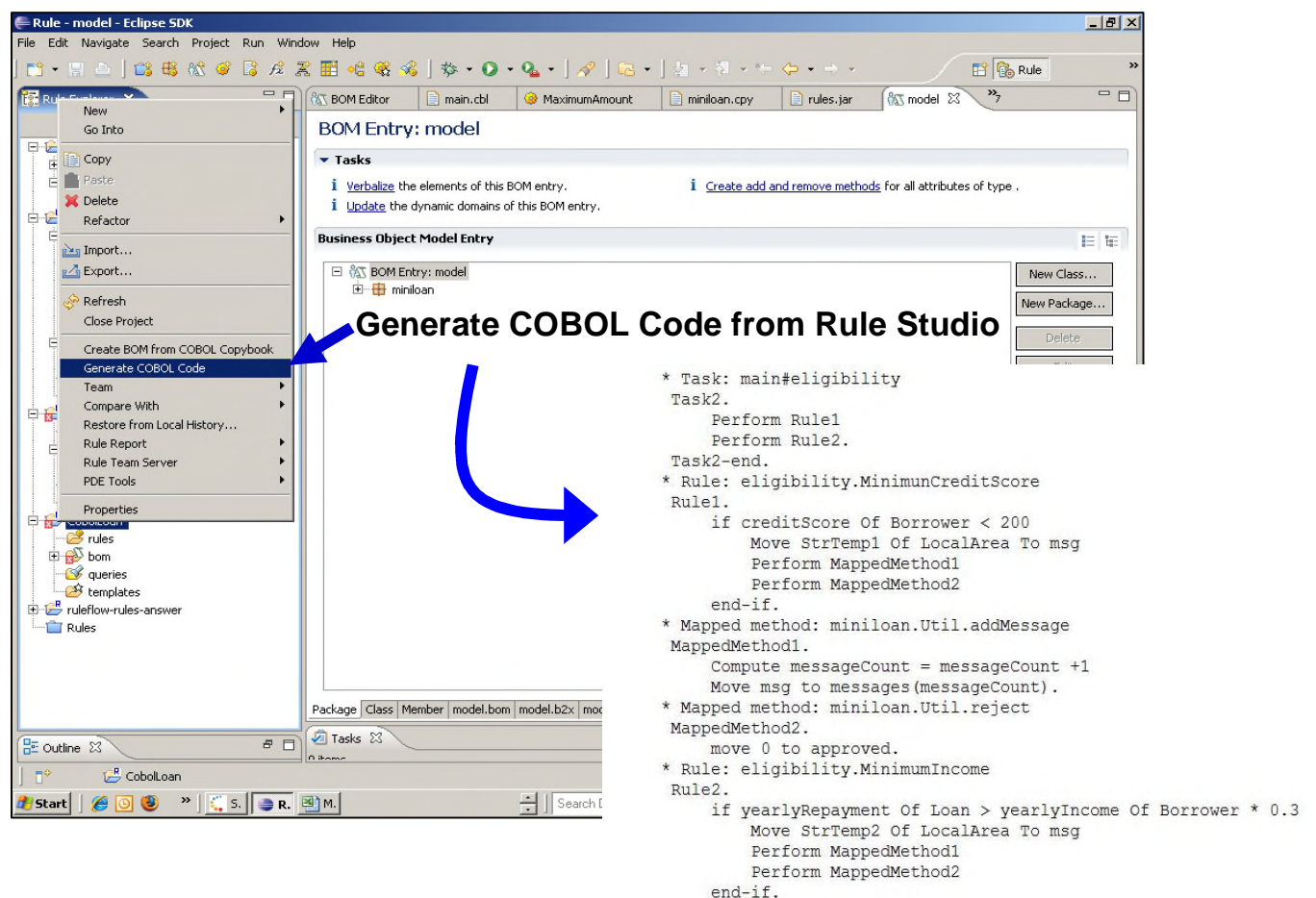


Fig. 3: Example of generating COBOL subprogram in ILOG Rule Studio

When you have finished defining the rule flow and authoring the business rules, you are ready to generate COBOL code from these rule artifacts. The ILOG Rules for COBOL generator takes the rules, decision tables and decision trees from the rule project and generates the COBOL code in the order given in the rule flow. The code is generated as a COBOL subprogram and does not require additional modifications or editing. Compatible with IBM Enterprise COBOL, the subprogram can be compiled and linked with a main program on the mainframe.

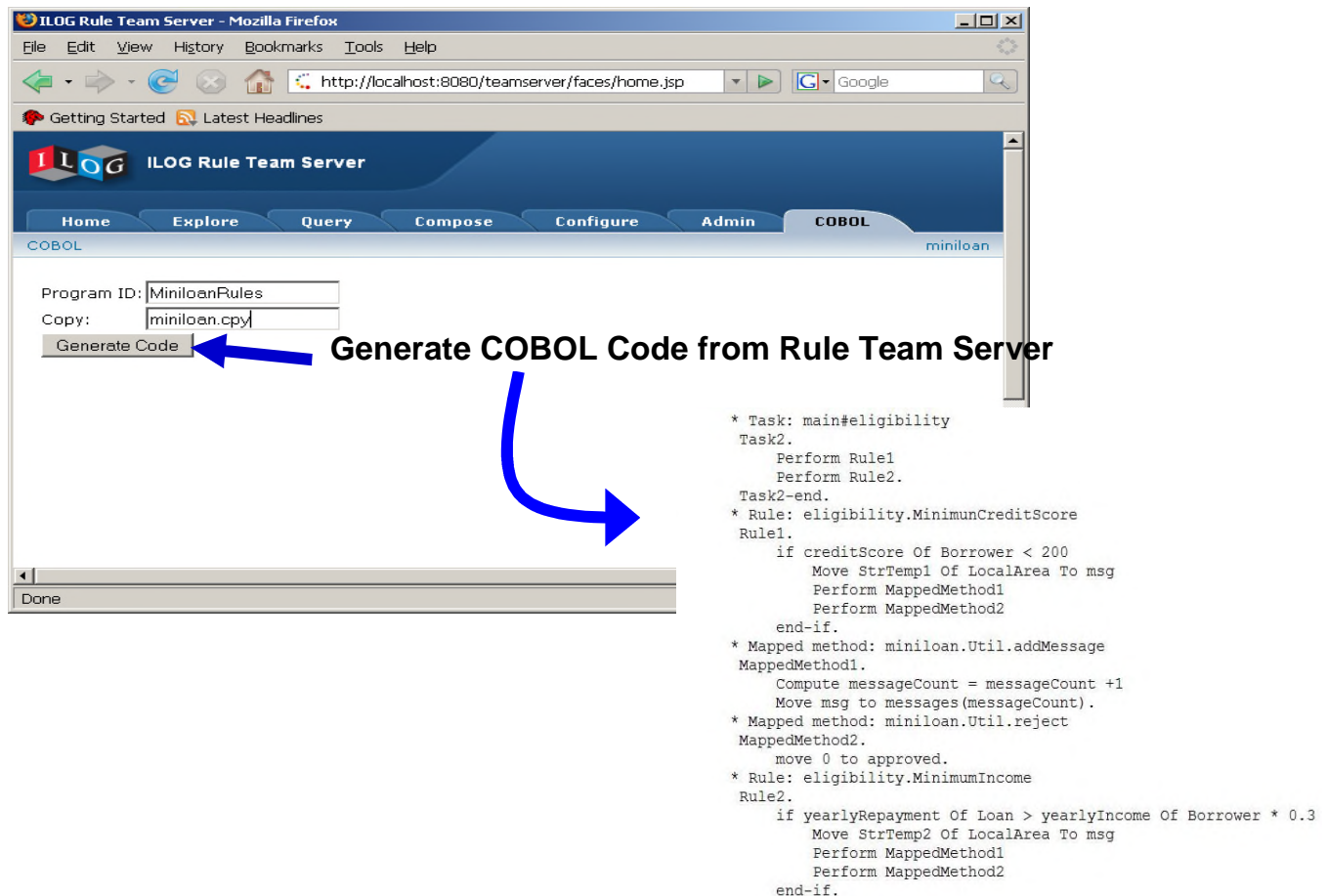


Fig. 4: ILOG Rules for COBOL generates COBOL code for insertion into applications

### 3. Business Rule Management for COBOL Applications

To succeed in a rapidly changing business world, mainframe COBOL applications must be flexible and reliable and meet rapid return on investment (ROI) expectations. Many traditional COBOL applications fall short of these requirements.

With ILOG Rules for COBOL, you can deploy business rule applications onto your mainframe. Your organization can incrementally move business rules within their COBOL application code to the BRMS, where they can be externally managed. Several customers have realized the benefits of using our BRMS with their COBOL applications in just two weeks. They started with a few decisions identified in key business processes, externalized their rules and managed them in ILOG JRules and then generated COBOL code for execution in their mainframe applications.

Generally, our customers have two reasons for using ILOG Rules for COBOL:

1. **Maintaining existing COBOL applications** – They want to make their mainframe applications more agile without having to completely reengineer large business applications and better manage business policy without changing technologies.
2. **Migrating their COBOL applications over time** – ILOG Rules for COBOL provides the ability to share business rules between COBOL and Java applications to ensure they produce the same results. Organizations migrating to Java can begin to incrementally move their business processes or business decisions within their business applications – one at a time without a large disruption.

Both options provide the same business management benefits with the use of ILOG JRules as a full-featured BRMS and ILOG Rules for COBOL to author and manage rules and generate COBOL code for mainframe COBOL applications.



The ILOG JRules rule repository addresses the specific needs of rule-based policy management with:

1. **Complete multi-project rule management** – The ILOG Rule Team Server repository is an enterprise repository, storing in a single place multiple independent or dependent rule projects and their histories.
2. **Multiuser access** – The repository supports automatic rule-level locking, in addition to user-managed persistent locks.
3. **Scalability** – The repository scales to dozens of users working on the same or different projects and hundreds of thousands of individual rule artifacts.
4. **Full version and history of rule artifacts** – The repository serves as a fully versioned content management system for BOM, vocabulary and rule artifacts.
5. **Baseline management** – The repository maintains “baselines” of rule project states, helping any previous project state for which a baseline has been created to be recalled for examination. A project state can be “rolled back” to any previous baseline.

The BRMS can also serve as the bridge between your mainframe and SOA strategy. The business rules managed within the BRMS and used to generate COBOL code can be used as business rules in Java applications and to create decision services for your SOA strategy.

By using the BRMS with your COBOL applications, your organization can realize the following bottom-line benefits:

- Preserved business value of existing assets and rules managed as intellectual assets in the BRMS
- Minimized business risk by ensuring consistency of the business rules throughout business applications
- Improved competitive position through enhanced business service and reuse with the BRMS technology
- Enhanced cost reduction and ROI with reduced total cost of ownership

## 4. Conclusion

With IBM ILOG JRules and ILOG Rules for COBOL, you can externalize and manage the business rules in your organization’s COBOL applications without disruption or a major development project. Furthermore, you can use the externalized rules as decision services within an SOA strategy.

With ILOG JRules and ILOG Rules for COBOL, you can:

- Maximize the benefits of your commitment to COBOL
- Execute natively using existing application flows
- Use rules in COBOL now and Java and SOA later