



Databorough

Visual Guide to Business Rules Reuse

Steve Kilner

Developers of legacy AS/400 systems have generally been unable to reuse their business rules effectively because of the monolithic nature of most legacy RPG applications.

Some of the benefits of better business rules reuse are:

- Facilitates analysis and portability of business logic
- Makes business logic more accessible to end users and business analysts
- Lowers the cost of developing and modifying business logic
- Externalizes rules so they can be shared by multiple applications
- Improves architecture by further separating UI, database, and business logic layers

This guide has the following sections:

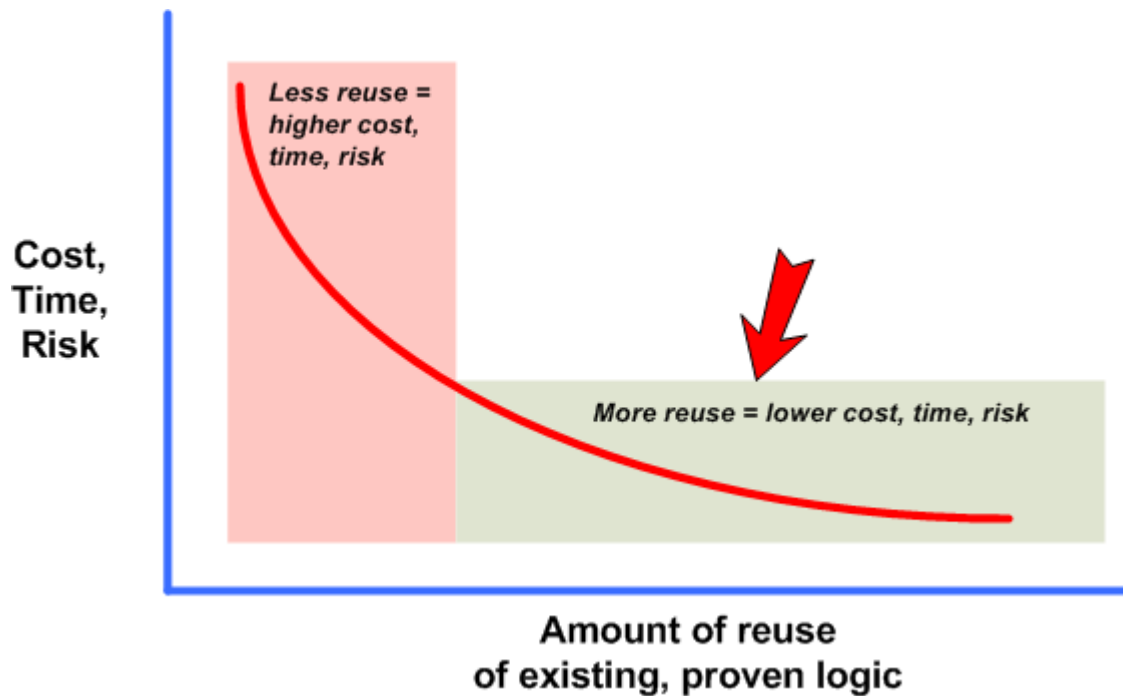
- [The key Concept for Managers of Legacy Systems](#)
- [How Legacy IT Managers Should Think About Business Rules](#)
- [Anatomy of a Business Rule](#)
- [Typical Reasons for Extracting Business Rules](#)
- [Using Business Rules For Reengineering and Rewrite Projects](#)
- [Using Business Rules For Projects Replacing Legacy Systems with Purchased Packages](#)
- [Using Business Rules For Maintenance Activities](#)
- [Detailed Information About X-Analysis and Business Rules](#)

This late in the game approach, however, misses out on opportunities to reduce application management costs, improve maintenance results and increase an application's value throughout its life-cycle. The overall benefits are:

- To help Business Analysts to understand and work with the system
- To communicate system functionality to users and management
- To help developers understand existing functionality, what it is and where it is located
- To improve system quality by facilitating consistency and accuracy
- To increase programmer productivity by enabling code reuse
- To feed the process of reengineering and migration
- To evaluate the suitability of packaged software being considered for purchase

The Key Concept for Managers of Legacy Systems

What is the single most critical factor in modernization, reengineering and rewrite projects?



By recovering existing, proven business logic that represents years of investment and development, IT organizations lower cost, time and risk for all types of projects that enhance or replace their legacy applications:

- Reengineering and rewriting projects
- Replacement projects that deploy purchased off the shelf packages
- Ongoing maintenance and enhancement

The more design information you recover, in the form of business rules and data model rules, the more successful you will be at lowering your costs, time lines and risks for such projects.

Here's a critical truth:

The single most cost-effective and accurate source of business requirements is always your legacy application.

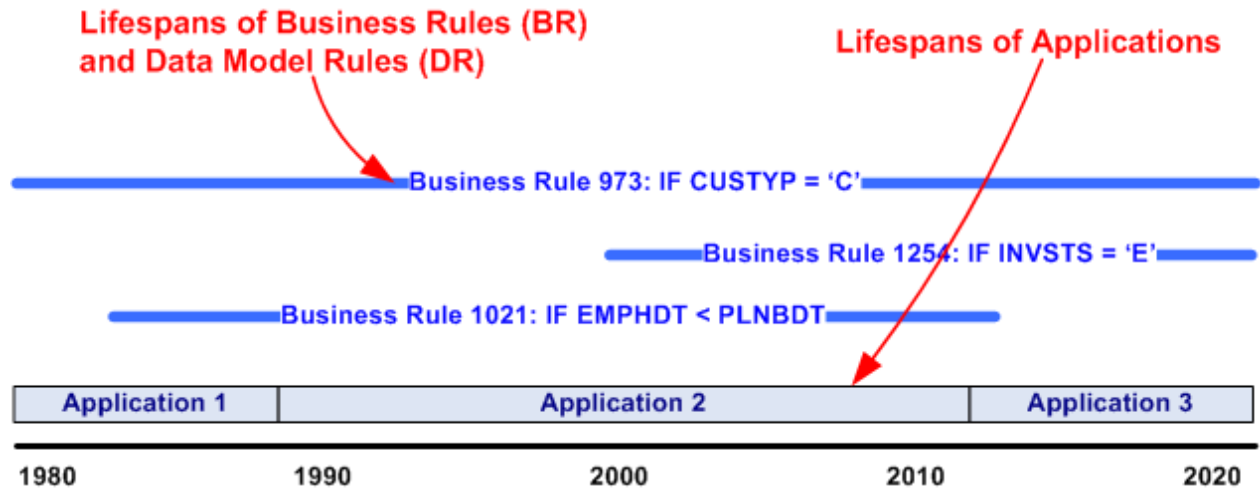
How Legacy IT Managers Should Think About Business Rules

Having the ability to look at applications from a Business Rule perspective is critical for this reason:

It is the only perspective that survives all applications, all languages, and all platforms, over time, for an organization.

You could rewrite your RPG IBM i application to any other language on any other platform, and still have the same business rules.

The following diagram illustrates this point:



You can probably think of business rules in your own organization that have been in place almost since the beginning of the organization, possibly before computers were even used. For example, "if customer type is a government agency give a 10% discount", or "if quantity on hand falls below the reorder point, create a purchase order."

Business Rules and Data Model Rules span the lifetimes of multiple systems because they are business constructs, not system constructs.

A second critical reason for adopting a business rule perspective is implied in that statement: business rules are defined by, and belong to, the business - surely they should be able to read and understand them.

Example

Here is a section of code implementing a business rule, as seen in RPG:

```

0261.00      C      SPERSON      CHAIN      SLMEN
0262.00      C      *IN99        IFEQ      '1'
0263.00      C      MOVE        '1'          *IN34
0264.00      C      MOVE        'OEM0023'    MSGID
0265.00      C      CALL        RTNMTX
0266.00      C      PARM        MSGID          7
0267.00      C      PARM        ERRMSG        132
0268.00      C      MOVE        *ALL'- '     PNAME
0269.00      C      LEAVE
0270.00      C      ENDIF
    
```

Here is that same logic expressed as a business rule, in a form that both users and analysts can understand.

```
Rule 71 Rep not found on Salespersons
  IF Not Found(Salespersons)
    ERROR 'Salesperson not found'
    Full_Name = *ALL'-'
  END
```

Because business rules belong to the users they should be understandable by the users.

Anatomy of a Business Rule

The above example illustrates two key points about recovering business rules from a legacy application:

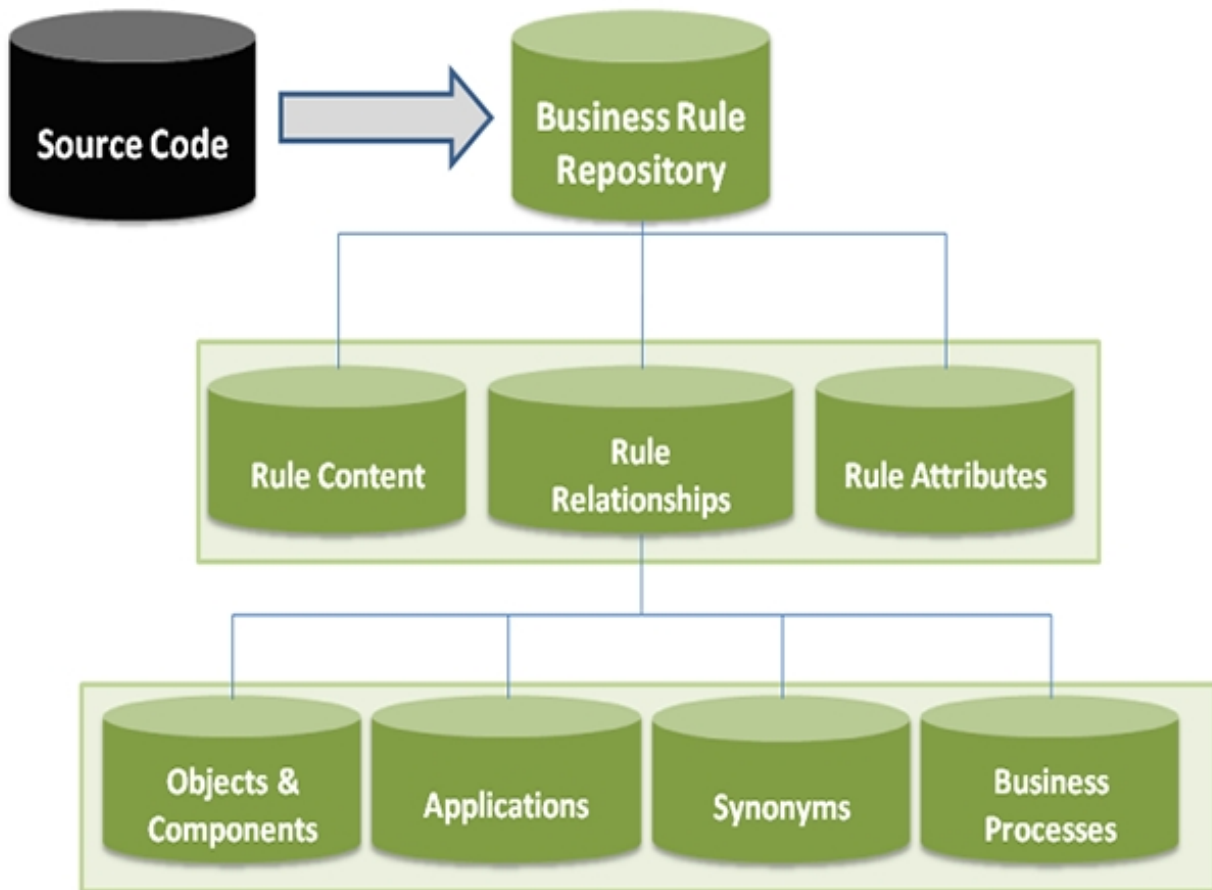
- How to identify business rules
- How to translate business rules into a readable form

X-Analysis follows a straightforward but internally complex algorithm for identifying business rules: if a conditional piece of logic is conditioned by a variable that is in any way related to a database field, then it is a business rule. This may sound simple enough on the surface, but the challenge is to track down fields that are remotely connected to database fields; the association may be through a series of MOVE or EVAL operations, through parameter passing, through renames and so on. By making use of its extensive internal cross reference capabilities X-Analysis is able to easily accomplish this task.

Similarly X-Analysis builds on its extensive pseudocode capabilities to translate program logic into human readable logic.

Business rules are extracted by reading through source and object data and storing the rules in a repository. The rules in the repository are stored in standard DB2 tables and may be viewed in client GUI or exported to Word, Excel or XML. They may also be processed by your own custom programs if you want to feed other tools or BRMS's such as iLog or JBoss Drools.

The logical structure of the X-Analysis business rule repository is shown by the following diagram:



Rule Content - consists of two parts:

- The rule itself, as a logic condition
- The processing logic that is executed if the condition is true

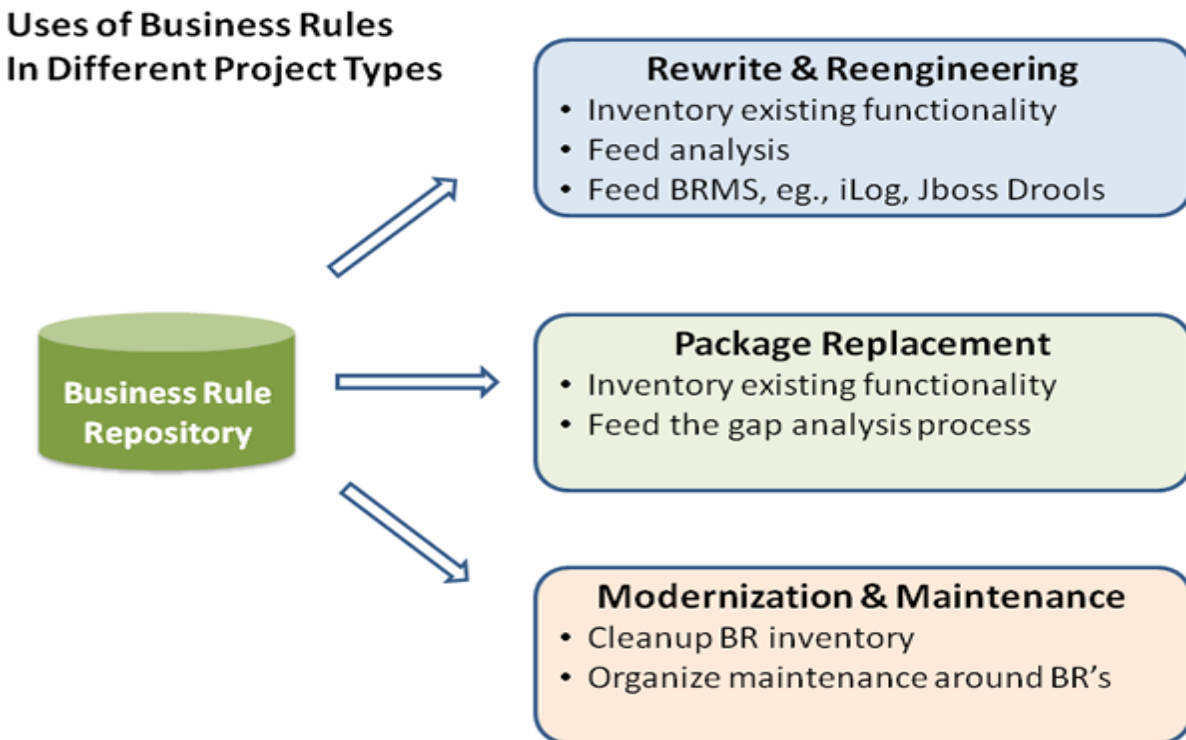
Rule Relationships - rules can have up to four types of relationships:

- To objects and components, such as programs, files, screens and fields
- To applications, if you subdivide your application into application areas
- To synonyms, as in other fields that are related to the primary field associated with the rule
- To business processes, typically expressed in the form of UML Activity Diagrams

Rule Attributes – the rule attributes describe the interface to the rule, sometimes referred to as “facts”; in other words, the data requirements for the rule both as input parameters and output parameters.

Typical Reasons for Extracting Business Rules

The uses of business rules is dependent on the nature of the work being done. The following diagram shows a high level view of business rule uses for different types of projects:



A fundamental concept that legacy system managers should understand is that reusing their existing business logic is one the best values available in IT.

To not reuse and recycle time-tested, proven business logic is to waste the huge investment the organization has made in getting the existing system to meet business needs over the years.

30,000 Rules

As a partner to many AS/400-IBM i shops around the world over the last 25 years, Databorough has had the opportunity to analyze many different applications. A "typical" RPG application consists of around a million lines of code (though some are much, much larger) and contains around 30,000 business rules.

If you are going to rewrite your application, or replace it with a package solution, you would need to define the business rules required for the new application. How do you define all those rules?

Imagine yourself in the following conversation:



Hi, this is Steve in IT, do you have time to define 30,000 business rules with me?

Actually, they're mostly the same rules we have now.



mmmm...

Uh, why don't just use those, ok?

Even the users get it!

Key Point

To not reuse and recycle time-tested, proven business logic is to waste the huge investment the organization has made in getting the existing system to meet business needs over the years.

For legacy systems owners it is during the transition to new or rewritten applications that extracting business rules represents the greatest possible value. are most commonly extracted and used to facilitate the process and lower both transition costs and risks.

Using Business Rules For Reengineering and Rewrite Projects

As the above "phone conversation" shows, it's wasteful of time and money to start off new rewrite projects by attempting to define all the business rules from scratch. In most cases, the majority of existing business rules in the legacy system will be carried forward to the new system. That's not to say there won't be additions and changes, but typically most of the existing business rules remain valid and useful.

There are three general approaches to reusing the recovered business rules:

- Feed the recovered rules into a modeling tool such as the IBM Rational toolset or any other UML/XML based tool
- Feed the recovered rules into a Business Rule Management System (BRMS), such as iLog or JBoss Drools
- Feed the recovered rules to analysts, users and developers manually through the use of the X-Analysis GUI and generated Word, Excel and XML documents.

This greatly facilitates analyst development of specifications for the new system, reducing time and cost, and improving communication and accuracy.

Using Business Rules For Projects Replacing Legacy Systems with Purchased Packages

The primary challenge of replacing legacy systems with purchased Commercial Off The Shelf (COTS) packages is understanding the differences, or gaps, between the systems. While IT and users are obviously more familiar with their own legacy system, they typically do not have a complete inventory of all of its business logic, which the business is dependent upon for operations and management.

Using a tool like X-Analysis to systematically recover all the business rules, as well as the data model rules, from the legacy system provides a solid and complete foundation from which to evaluate gaps with the COTS package.

The recovered business rules and data model rules are useful in two phases of COTS projects:

- Doing the gap analysis to aid in selecting a package for purchases; i.e., vendor comparison
- Defining custom programming and modifications that need to be made for the selected package in order to meet the unique needs of the organization

Using Business Rules For Maintenance Activities

As should be apparent from some of the earlier discussion in this guide, business rules are fundamental, sustained entity for an organization and it makes sense for IT to organize its efforts around such entities.

This is why you see many IT moving to implementing business logic through the use of Business Rule Management Systems. Such systems are more portable, durable, accessible and easier to maintain.

IT organizations that support legacy systems are well served to begin a transformation in how they organize their efforts so that they are in line with modern software practices. An excellent beginning is to take the following steps:

- Recover you business rules from your legacy systems
- Clean them up, by identifying and refactoring duplicate, obsolete or inconsistent rules

- Build a simple working repository of business rules that analysts, users and developers can work with, and that serves as the basis for all future development and enhancements

The overall benefits of adopting a business rule perspective for software maintenance include:

- To help Business Analysts better understand and work with the system
- To communicate system functionality to users and management
- To help developers understand existing functionality, what it is and where it is located
- To improve system quality by facilitating consistency and accuracy
- To increase programmer productivity by enabling code reuse
- To eventually feed the process of reengineering and migration
- To eventually evaluate the suitability of packaged software being considered for purchase

A repository of Business Rules should do more than simply be another system reference. It should be used to shape IT processes and communications. As noted in the introduction, Business Rules survive multiple iterations of application lifecycles and platform installations.

Businesses have business processes which are typically represented by what we call applications. Applications may be subdivided into sub-applications. Individual actions occur in applications and are typically comprised of functional business rules. Data model business rules define the data used within applications.

Detailed Information About X-Analysis and Business Rules

The traditional means of extracting business rules has been for business analysts to sit down with developers and walk through the code, often driven by existing documentation into which the business rules are mapped.

With a tool such as X-Analysis, however, new opportunities open up due to the ease and speed with which business rules can be extracted.

X-Analysis allows you to extract your business rules with just a few simple clicks in a matter of seconds.

Screenshots

Business Rules List

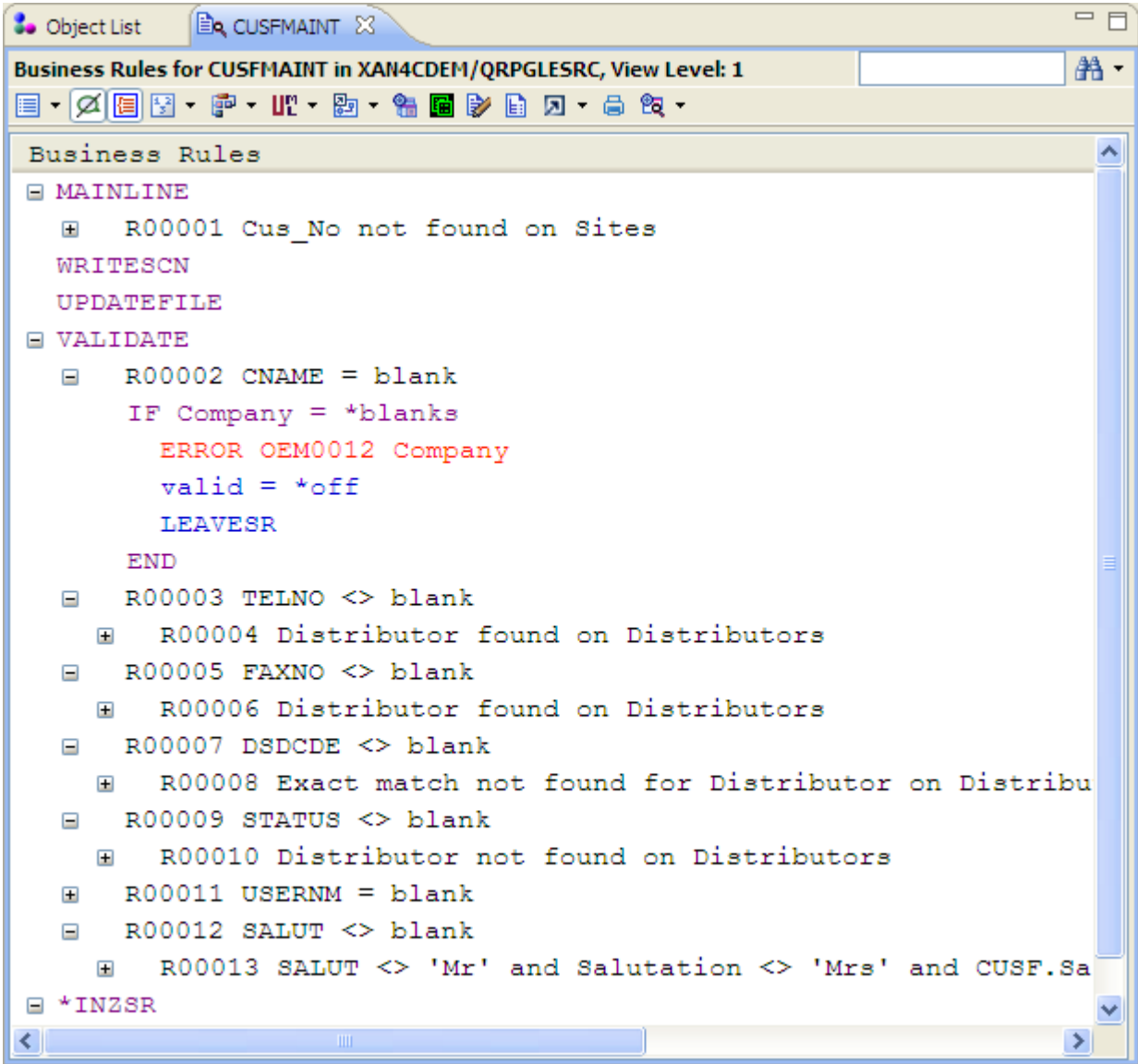
X-Analysis recovers the thousands of business rules in your system and provides them in a variety of different lists which can be sorted, filtered and exported.

Business Rules for *ALL, Number of Lines: 321

Source Member	Rule No.	Field	File	Rule
CBCONDET	00001	XWBCCD	CUSTS	Debtor not found on Purchases
CBCONDET	00002	PERSON	SLMEN	Rep not found on Salespersons
CBCONDET	00003	XWSTAT	ORDSTS	Status not found on Order_status_description
CBCONDET	00004	XWORDN	CONDET	XWORDN OF CONDET-RECORD <> LS-XWORDN
CBCONDET	00005	XWORDN	CONHDR	Contract found on Contract_Header
CBCONDET	00006	XWORDN	CONDET	XWORDN OF ZZFT01-O = ZEROS
CBCONDET	00007	XWORDN	CONDET	Exact match not found for Contract on Contract_Detail
CBCONDET	00008	XWBCCD	CUSTS	Debtor not found on Purchases
CBCONDET	00009	XWSTAT	ORDSTS	Status not found on Order_status_description
CBCONDET	00010	PERSON	SLMEN	Rep not found on Salespersons
CBCONDET	00011	XWORDN	CONHDR	Contract not found on Contract_Header
CBCONDET	00012	XWORDN	CONDET	Contract not found on Contract_Detail
CBCONDET	00013	XWORDN	CONDET	Contract found on Contract_Detail
CBCONDET	00001	XWBCCD	CUSTS	Debtor not found on Purchases
CBCONDET	00002	PERSON	SLMEN	Rep not found on Salespersons
CBCONDET	00003	XWSTAT	ORDSTS	Status not found on Order_status_description
CBCONDET	00004	XWORDN	CONDET	CON-CONTRACT-NUMBER OF CONDET-RECORD <> LS-XWORDN
CBCONDET	00005	XWORDN	CONHDR	Contract found on Contract_Header
CBCONDET	00006	XWORDN	CONDET	XWORDN OF ZZFT01-O = ZEROS
CBCONDET	00007	XWORDN	CONDET	Exact match not found for Contract on Contract_Detail
CBCONDET	00008	XWBCCD	CUSTS	Debtor not found on Purchases
CBCONDET	00009	XWSTAT	ORDSTS	Status not found on Order_status_description
CBCONDET	00010	PERSON	SLMEN	Rep not found on Salespersons
CBCONDET	00011	XWORDN	CONHDR	Contract not found on Contract_Header
CBCONDET	00012	XWORDN	CONDET	Contract not found on Contract_Detail
CBCONDET	00013	XWORDN	CONDET	Contract found on Contract_Detail
CBCONHDR	00001	XWBCCD	CONHDR	XWBCCD OF CONHDR1-RECORD <> LS-SWBCCD

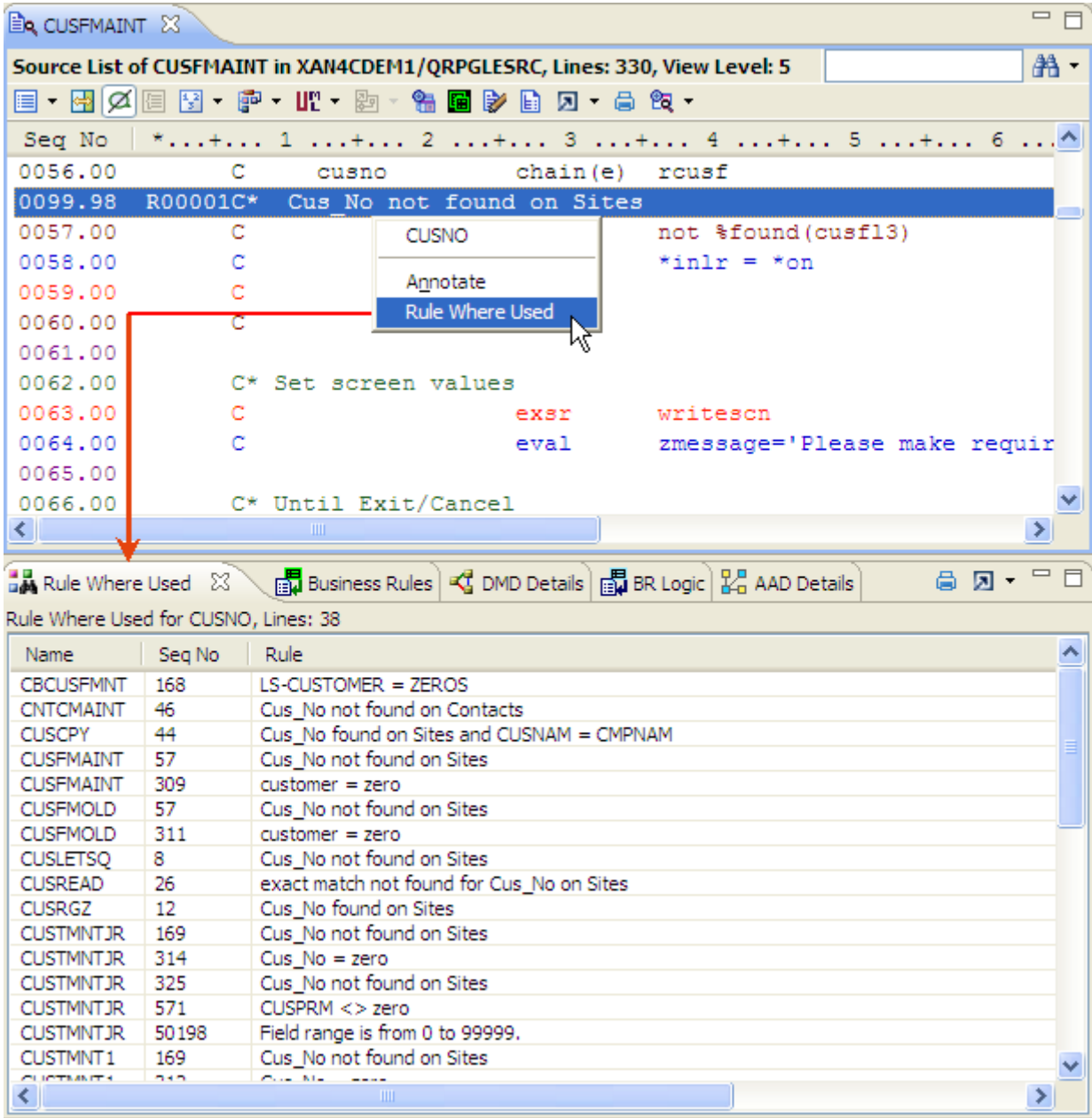
Business Rules Code

X-Analysis displays the business rules in your code as it is organized in your programs, by subroutines and procedures. You can instantly navigate between the business rule view of pseudocode and your RPG code.



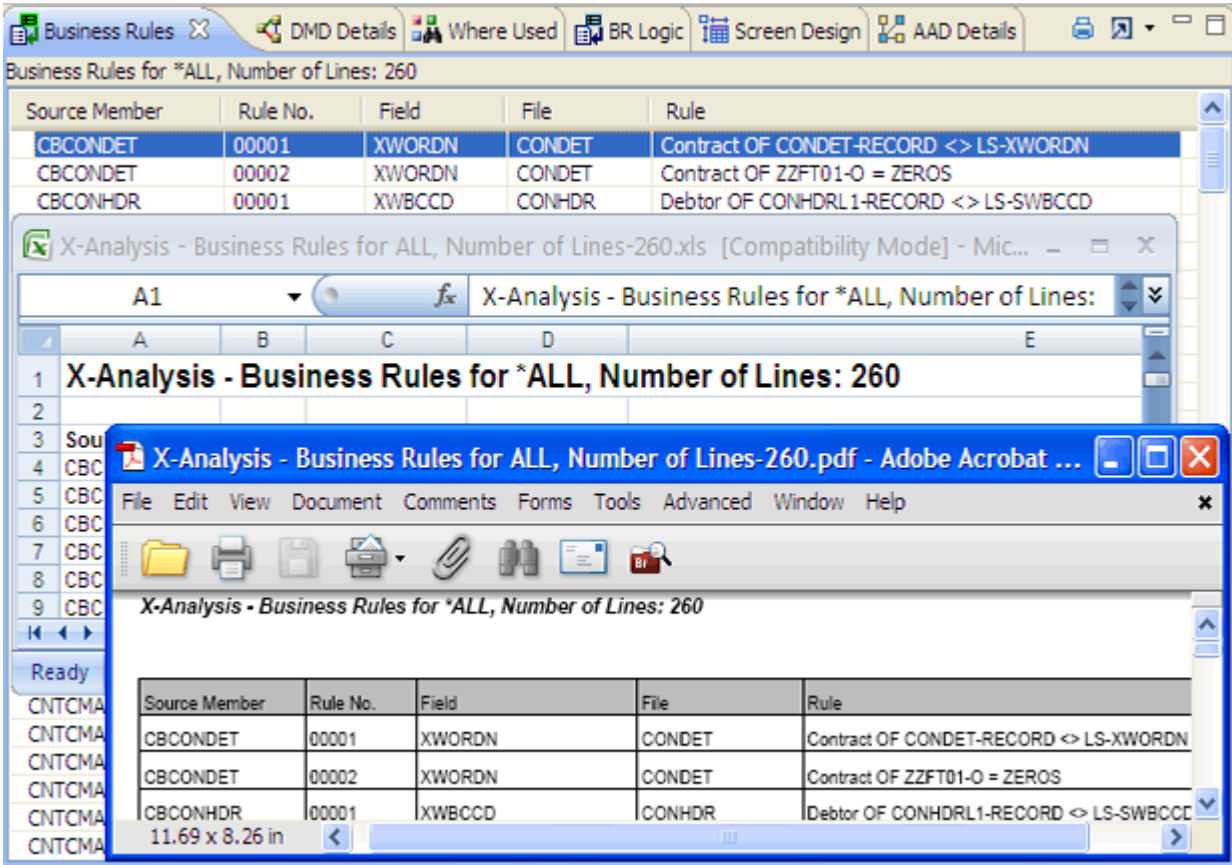
Business Rules Where Used

The Where Used facility enables you to see where a given variable is used to define business rules, or where similar or duplicate business rules can be found.



Export Business Rules

The export facility enables you to export business rules as Word, Excel or PDF documents.



Arrange for a Demonstration or Free Trial

[Contact us at Databorough](#) to arrange for a demonstration or free trial of X-Analysis for recovering business rules, data models, and application reengineering.