

Business Rule Extraction from COBOL applications

Enterprise Applications typically contain substantial amounts of Business Knowledge, which have been constructed, enhanced and tuned over many years. Typically such Legacy Enterprise Applications have little or no up-to-date documentation, and original authors and designers are no longer able or available to re-document the system.

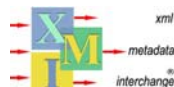
Manual re-documentation of such code is often a lengthy, hugely expensive and error-prone process.

SoftwareMining's BRE Toolkit uses sophisticated search algorithms to sieve through huge amount of code and isolate the Business Related code. The system allows assignment of descriptions in natural language to the set of variables referenced within the identified rule. This feature can be used to reproduce Business Rules in a Natural Language form (i.e. English).

One of the main features of the Tool is the ability to apply SoftwareMining's Language Translation Technology to the extracted Business Rules. The Rules can thus be exported as Java, C# or COBOL code snippets, ready for incorporation onto newly built applications.

Benefits

- Ease Application Maintenance
- Substantial gains in Speed and Accuracy for Business Rule Extraction processes
- Opening the system to Business-Analysis: GAP Analysis
- Business Rules exported as HTML - readily accessible to team members
- Application Flowcharting exported as SVG/HTML
- Re-architecture: Multi language & SOA
- Multi-User facility



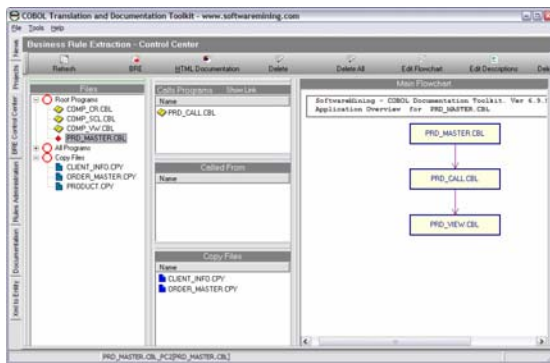
Code Inventory & Documentation

The system is allow easy maintenance of the Application, providing easy access to information such as:

- What are the entry points to the applications (Root Programs)
- What are the dependencies between program modules
- What are the dependencies on copy files
- What are data-structures are passed between calls, and how are they transformed.

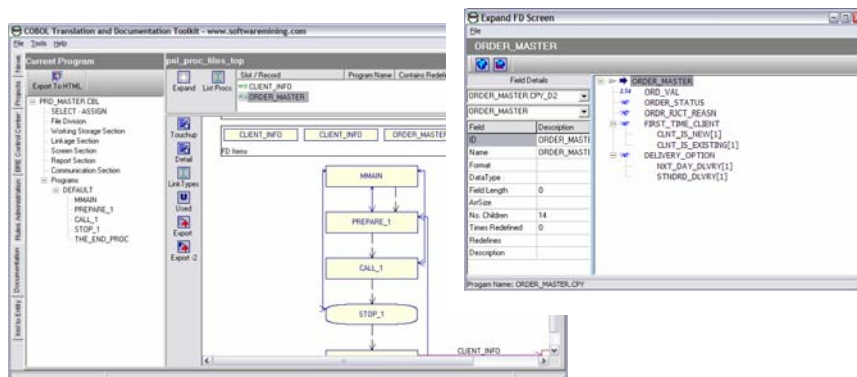
The main screen in BRE Toolkit provides an overview of the application, including:

- list of all programs, which copy files they reference and what other programs they call
- Copy Files - which programs references them
- Inter-program flowcharts.



The program flowcharts and Data-Division hierarchies are appropriately documented.

The documentation can be saved in HTML/SVG format.

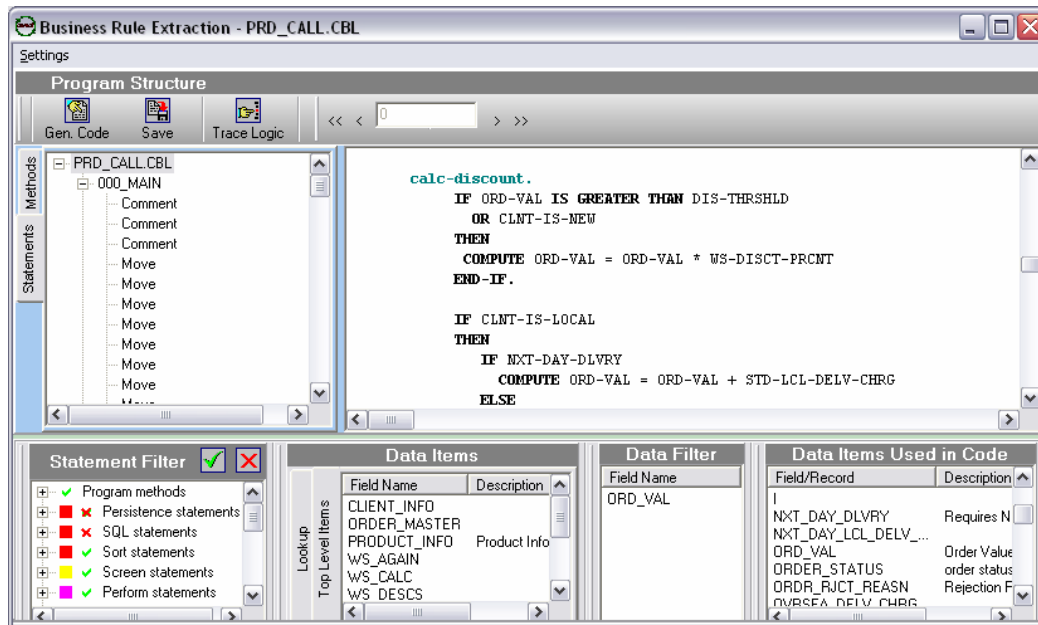


Business Rule Extraction

Most COBOL programs perform a variety of tasks. Eg. An Order-Processing program may calculate invoice, update inventory records, have data-entry screens etc. SoftwareMining's Business Rule Extraction Strategy works by providing Data and Statement-Category filters, and Advanced Search Algorithms to extract the relevant information from the programs.

All the Business Analyst / developers have to do, to start the business rule extraction process, is to:

- Specify a Variables Of Interest; telling the system to find all explicit / implicit (Thru redefines, and hierarchies) References to the Variable of Interest.
- Select Statement filters; For example, and a rule interested in calculation of Order-Values may not be interested in screen-handling, database handling, statements.



The application of the above filters results in identification of code addressing Variable of Interest, as well as all the other variables *referenced within the listed code segment*.

The following code snippet is produced as a result of business rule extraction on "Order Value" (variable ORD-VAL) on COBOL program

Business Rule Code

```
IF ORD-VAL IS GREATER THAN DIS-THRSHLD
  OR CLNT-IS-NEW
THEN
  COMPUTE ORD-VAL = ORD-VAL * WS-DISCT-PRCNT
END-IF .

IF CLNT-IS-LOCAL
THEN
  IF NXT-DAY-DLVRY
    COMPUTE ORD-VAL = ORD-VAL + STD-LCL-DELV-CHRG
  ELSE
    COMPUTE ORD-VAL = ORD-VAL + NXT-DAY-LCL-DELV-CHRG
  END-IF
ELSE
  COMPUTE ORD-VAL = ORD-VAL + OVRSEA-DELV-CHRG
END-IF .
```

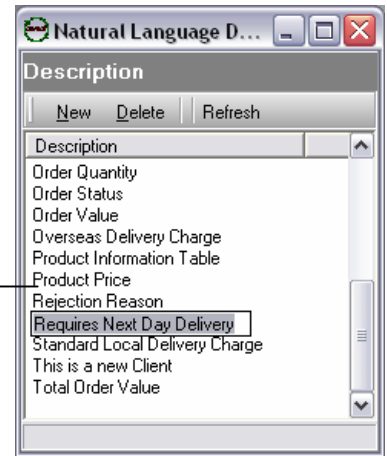
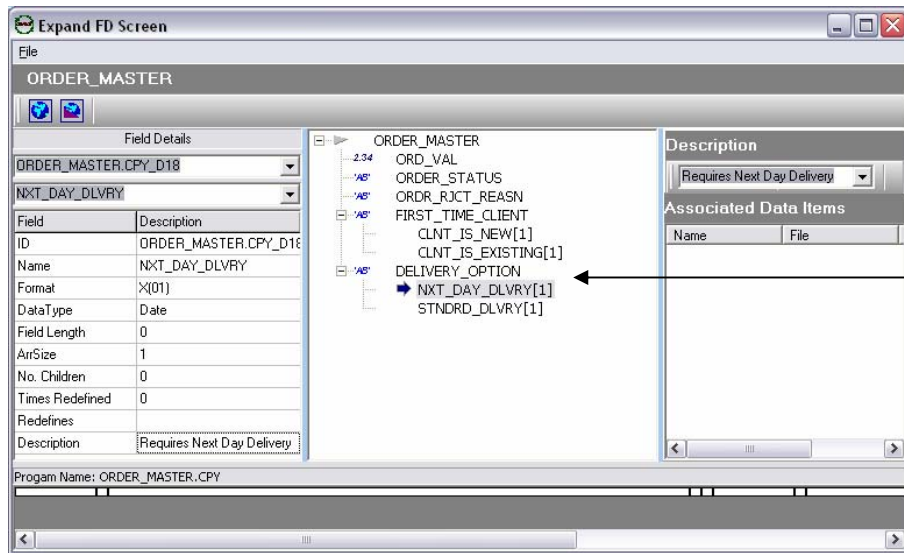
Translation of COBOL Rules to Natural Language

SoftwareMining's Business Rule Extraction Toolkit also facilitates the association of Natural Language Description with variables names. The descriptions are used, together with in-built heuristic based Natural Language substitution algorithms, to generate specifications in Pseudo-English from the extracted business rule. For example:

Business Rule

```
IF Order Value IS GREATER THAN lower discount limit
  OR This is a new Client
THEN
  Order Value = Order Value * discount percentage
END-IF .

IF client has local geographical location
THEN
  IF Requires Next Day Delivery
    Order Value = Order Value + Local Delivery Charge
  ELSE
    Order Value = Order Value + next day delivery charge
  END-IF
ELSE
  Order Value = Order Value + Overseas Delivery Charge
END-IF .
```



Use Drag-n-Drop to associate Variables and Descriptions

Translation of COBOL Rules to Java Rules

Often the aim of Business Rule Extraction is to produce specification for the rewrite of an application in Java or C#. SoftwareMining's COBOL to Java Translation system is an integrated part of the Business-Rule-Extraction Toolset. The translation module can be used to produce the business rules in **Maintainable and Legible** Java or C# code, together with all XML definitions for data-beans, Object-Relational Persistence Beans and the underlying framework.

```
Business Rule Code

/**
 * calcDiscount
 */

protected void calcDiscount (){
    if (orderMaster.getOrdVal() > iData.getDisThrshld() ||
        orderMaster.getFirstTimeClient()
            .equals(orderMaster.firstTimeClientClntIsNew))
    {
        orderMaster.setOrdVal(orderMaster.getOrdVal()
            * iData.getWsDisctPrct());
    }

    if (clientInfo.getLocalClnt()
        .equals(clientInfo.localClntClntIsLocal))
    {
        if (orderMaster.getDeliveryOption()
            .equals(orderMaster.deliveryOptionNxtDayDlvry))
        {
            orderMaster.setOrdVal(orderMaster.getOrdVal()
                + iData.getStdLclDelvChrg());
        } else
        {
            orderMaster.setOrdVal(orderMaster.getOrdVal()
                + iData.getNxtDayLclDelvChrg());
        }
    } else {
        orderMaster.setOrdVal(orderMaster.getOrdVal()
            + iData.getOvrseaDelvChrg());
    }
}
```

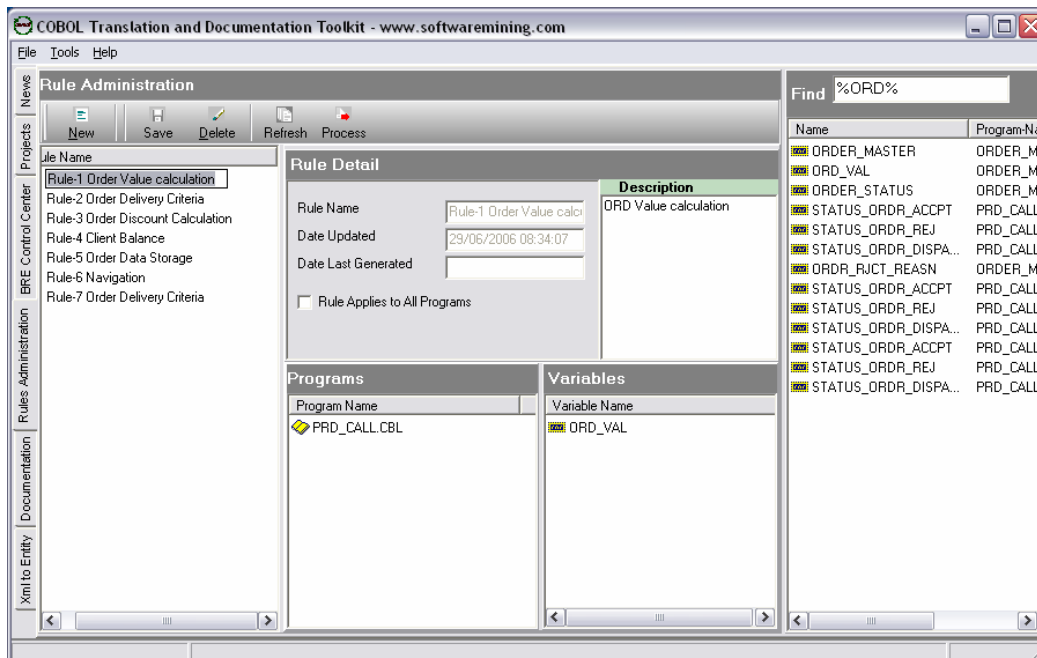
For further information please contact sales@softwaremining.com.

Rule Administration screen

Rule Administration screen is provides a central place for building and maintenance of rules.

An automatic documentation process can be applied to all the rules in this Screen.

The rules composed in this screen are independent of the application repository. I.e. They ease the re-generation of rules after code changes.



Dead Code Analysis

Legacy application often contain code and variable definitions longer utilized with the application.. SoftwareMining BRE toolset can help identify the dead code via a logical analysis of the application code. The technique traverses the application code (no data necessary) and identifies unreachable statements and methods and variables not addressed within the code. The system can then report its findings in a variety of ways, including writing the result to a color code HTML document.

Summary Information

Legacy application often contain code and variable definitions longer utilized with the application.. SoftwareMining BRE toolset can help identify the dead code via a logical analysis of the application code. The technique traverses the application code (no data necessary) and identifies unreachable statements and methods and variables not addressed within the code. The system can then report its findings in a variety of ways, including writing the result to a color code HTML document.

```
Business Rule
O100-OPEN-FILE.
  OPEN INPUT PARMFILE.
  IF NOT WS-STATUS-OK
    MOVE WS-ERROR-2 TO WS-ERROR-LINE
    GO TO Z100-FILE-ERROR
  ELSE
    PERFORM DISPLAY-AND-GO-BACK.
  MOVE 'Y' TO WS-FILE-OPEN-FLAG.
  MOVE WS-RECORD-STATUS TO L-PARML-FILE-STATUS.
```

Identifying Decision Paths

SoftwareMining's BRE Toolkit can allows automatic tracing through different paths that a program may take, creating a decision table for the main decisions a program takes. Take the following example:

```
Business Rule
IF Order Value > Client's Balance
  THEN
    MOVE Rejection Reason to order status
    MOVE 'Insufficient Credit' to Rejection Reason
  ELSE
    IF All Items In Stock
      THEN
        MOVE order dispatched to order status
        Client's Balance = Client's Balance - Order Value
      ELSE
        MOVE Rejection Reason to order status
        MOVE 'One or more items out of stock'
          to Rejection Reason
    END-IF
  END-IF.
```

The above code segment containing 3 simple decisions (IF/THEN/ELSE conditions). The BRE can automatically identify these to:

```
IF Order Value > Client's Balance
  THEN
```

```
MOVE Rejection Reason to order status
MOVE 'Insufficient Credit' to Rejection Reason
ELSE
END-IF.
```

```
IF Order Value > Client's Balance
THEN
ELSE
  IF All Items In Stock
  THEN
    MOVE order dispatched to order status
    Client's Balance = Client's Balance - Order Value
  ELSE
  END-IF
END-IF.
```

```
IF Order Value > Client's Balance
THEN
ELSE
  IF All Items In Stock
  THEN
  ELSE
    MOVE Rejection Reason to order status
    MOVE 'One or more items out of stock'
    to Rejection Reason
  END-IF
END-IF.
```

Due to the simplicity of the above code, the manual identification of the 3 decisions is fairly easy. However, the task becomes substantially more difficult for larger and more complex code typically found in live applications.

Features List

SoftwareMining's BRE Tool Features:

- Application Overview
- Application Wide, Program Connectivity Flowchart and Rule Composer
- Flowchart of Individual Programs
- Identifies COPY File usage throughout application
- Support for REDEFINES and Hierarchical definitions
- Support for GOTO, PERFORM and PERFORM-THRU
- Dead Code Identification
- Dead Variable identification
- "Decision Tree" Identification
- Generation of Application and Program Flowcharts
- HTML Documentation of Hierarchical COBOL structures
- Repository Based

Business Rules Functionality

- Extraction of Business Rules using "Variable" or "Statement" type filters
- Assignment of Descriptions to Variables
- Business Rules in COBOL, Natural Language (ENGLISH), Java or C#