Business Rules Extraction (BRX) with Application Analyser

A tool-assisted solution for identifying and documenting the critical business processes hidden in your COBOL applications.

Business rules and their importance

Business rules are the sections of code in your COBOL programs that provide the business logic and functionality for your application. Conditions, calculations and proprietary algorithms control how your application works and what it actually does for your organisation. Business rules represent the intellectual property of the organisation, providing competitive differentiation and business optimisation.

Business rules are deeply embedded in the existing COBOL program code and depending on how old applications are, many of these rules may not be known or documented anywhere outside of the code itself.

Over time, rules inevitably change as the underlying business processes evolve. In many organisations, it becomes impossible for anyone to fully understand all of the logic and conditions that are embedded in their COBOL applications.

This lack of understanding can become a problem when trying to enhance the application to meet new business needs. It becomes an even greater concern when organisations look to replace or redevelop an application. Requirement definitions involving changes to existing business rules or the addition of new rules will be on a shaky footing if the existing, already embedded rules are not well understood and documented. Projects will struggle to deliver the desired functionality and may even introduce unexpected processing errors into the changed application.

The challenge

Identifying business rules hidden within a COBOL application is an arduous task if undertaken manually. Manual approaches are also open to inaccuracy and omissions, leading to unreliable results. This drives up costs as post-delivery corrections are always expensive.

A software tool that can automatically identify business rules is very appealing. However, previous attempts to fully automate the extraction of business rules have led to mixed results, often generating a huge amount of data, much of which doesn't represent actual business rules.

We believe that the best approach to

Business Rules Extraction (BRX) with Application Analyser

identifying business rules hidden in an application is to combine the subject matter expertise of an application analyst or developer with a tool that provides an automated process for exposing potential business rules. This dramatically improves the productivity compared to a manual approach, while ensuring that only true business rules are extracted and documented.

Our solution

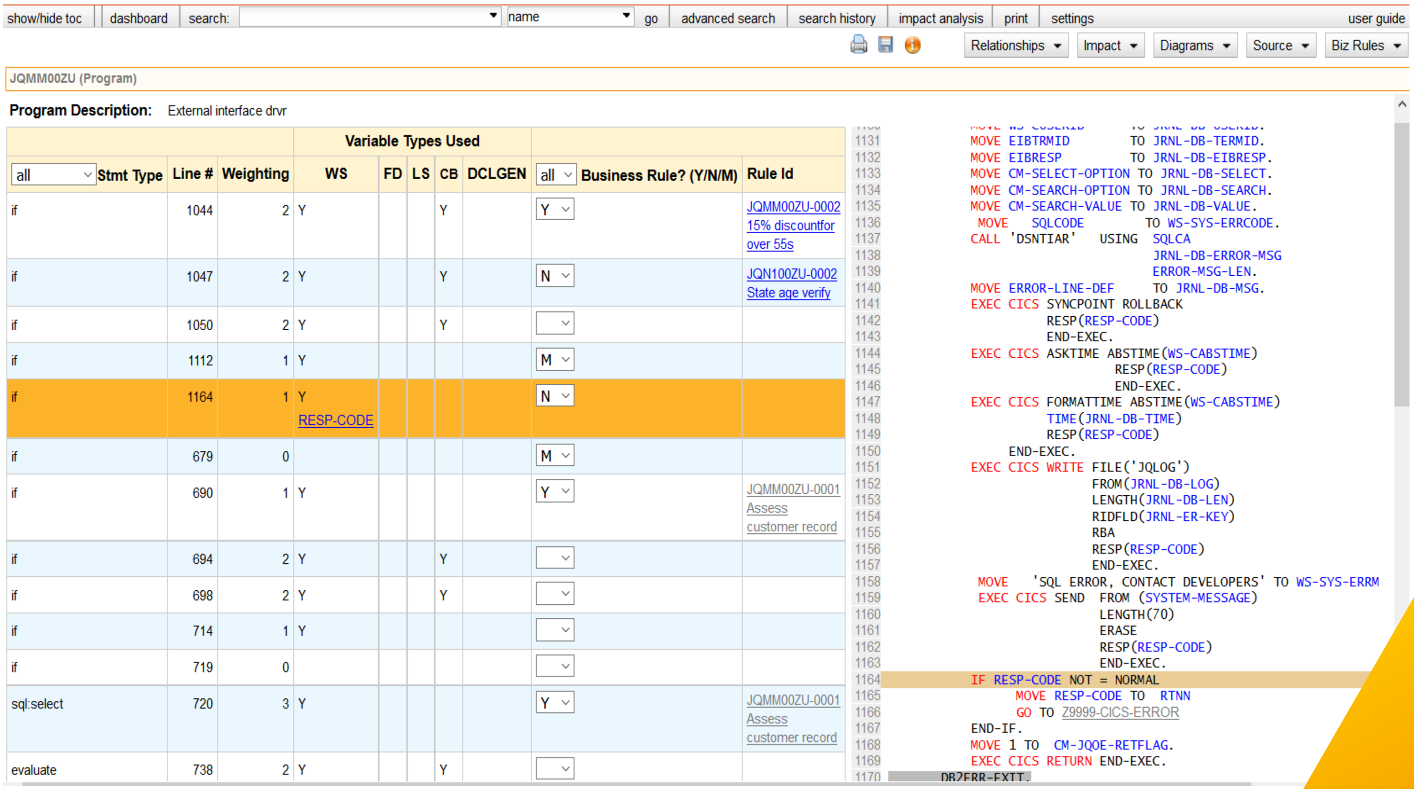
A heuristic, semi-automated process

The Business Rules Extraction (BRX) add-on module to our Application Analyser product leverages the deep semantic understanding already captured in the Application Analyser database. BRX provides user defined criteria to search for potential business rules in an application.

The criteria can be set for candidate statement types, as well as candidate data types within a statement. For example, an IF statement that is based on data derived from a database table or screen input is more likely to be a business rule, than an IF statement that is based on internal Working Storage data.

BRX identifies all candidate statements and then applies a weighting to them depending on the criteria created by the user. This quickly and automatically produces a list of candidate business rules that an expert application analyst can evaluate and subsequently accept or reject as a bona fide business rule. Each identified business rule can be documented in free text as well as code snippets. Rules and their descriptions are stored in the Application Analyser database.

Once business rules are indentified and documented, BRX can generate process flow models showing how transactions are processed through the application, depicting each program and their associated business rules. This is extremely helpful for developers enhancing applications or replicating the application's functionality in a replacement or re-development project.



JQMM00ZU (Program)

Program Description: External interface drvr

			Variable Types Used							
all **Stmt Type**	**Line #**	**Weighting**	**WS**	**FD**	**LS**	**CB**	**DCLGEN**	all **Business Rule? (Y/N/M)**	**Rule Id**
if	1044	2	Y			Y		Y	JQMM00ZU-0002 15% discountfor over 55s
if	1047	2	Y			Y		N	JQN100ZU-0002 State age verify
if	1050	2	Y			Y			
if	1112	1	Y					M	
if	1164	1	Y RESP-CODE					N	
if	679	0						M	
if	690	1	Y					Y	JQMM00ZU-0001 Assess customer record
if	694	2	Y			Y			
if	698	2	Y			Y			
if	714	1	Y						
if	719	0							
sql:select	720	3	Y					Y	JQMM00ZU-0001 Assess customer record
evaluate	738	2	Y			Y			

```
           MOVE EIBTRMID        TO JRNL-DB-TERMID.
           MOVE EIBRESP         TO JRNL-DB-EIBRESP.
           MOVE CM-SELECT-OPTION TO JRNL-DB-SELECT.
           MOVE CM-SEARCH-OPTION TO JRNL-DB-SEARCH.
           MOVE CM-SEARCH-VALUE TO JRNL-DB-VALUE.
            MOVE   SQLCODE        TO WS-SYS-ERRCODE.
           CALL 'DSNTIAR'   USING  SQLCA
                                   JRNL-DB-ERROR-MSG
                                   ERROR-MSG-LEN.
           MOVE ERROR-LINE-DEF     TO JRNL-DB-MSG.
           EXEC CICS SYNCPOINT ROLLBACK
                     RESP(RESP-CODE)
                     END-EXEC.
           EXEC CICS ASKTIME ABSTIME(WS-CABSTIME)
                              RESP(RESP-CODE)
                              END-EXEC.
           EXEC CICS FORMATTIME ABSTIME(WS-CABSTIME)
                     TIME(JRNL-DB-TIME)
                     RESP(RESP-CODE)
                END-EXEC.
           EXEC CICS WRITE FILE('JQLOG')
                           FROM(JRNL-DB-LOG)
                           LENGTH(JRNL-DB-LEN)
                           RIDFLD(JRNL-ER-KEY)
                           RBA
                           RESP(RESP-CODE)
                           END-EXEC.
            MOVE   'SQL ERROR, CONTACT DEVELOPERS' TO WS-SYS-ERRM
            EXEC CICS SEND  FROM (SYSTEM-MESSAGE)
                            LENGTH(70)
                            ERASE
                            RESP(RESP-CODE)
                            END-EXEC.
           IF RESP-CODE NOT = NORMAL
                 MOVE RESP-CODE TO  RTNN
                 GO TO Z9999-CICS-ERROR
           END-IF.
           MOVE 1 TO  CM-JQOE-RETFLAG.
           EXEC CICS RETURN END-EXEC.
       DB2ERR-EXIT.
```

Business Rules Extraction (BRX) with Application Analyser

Benefits

- > Identify the hidden business rules in your COBOL applications in a cost effective, highly productive and accurate manner
- > User configured to your business and application environments
- > Supports a wide variety of COBOL compilers
- > Document business rules in English
- > Display business rules in process flows, referencing underlying programs
- > BRX builds on the proven capabilities of our Application Analyser solution
- > Display business rules in process flow models

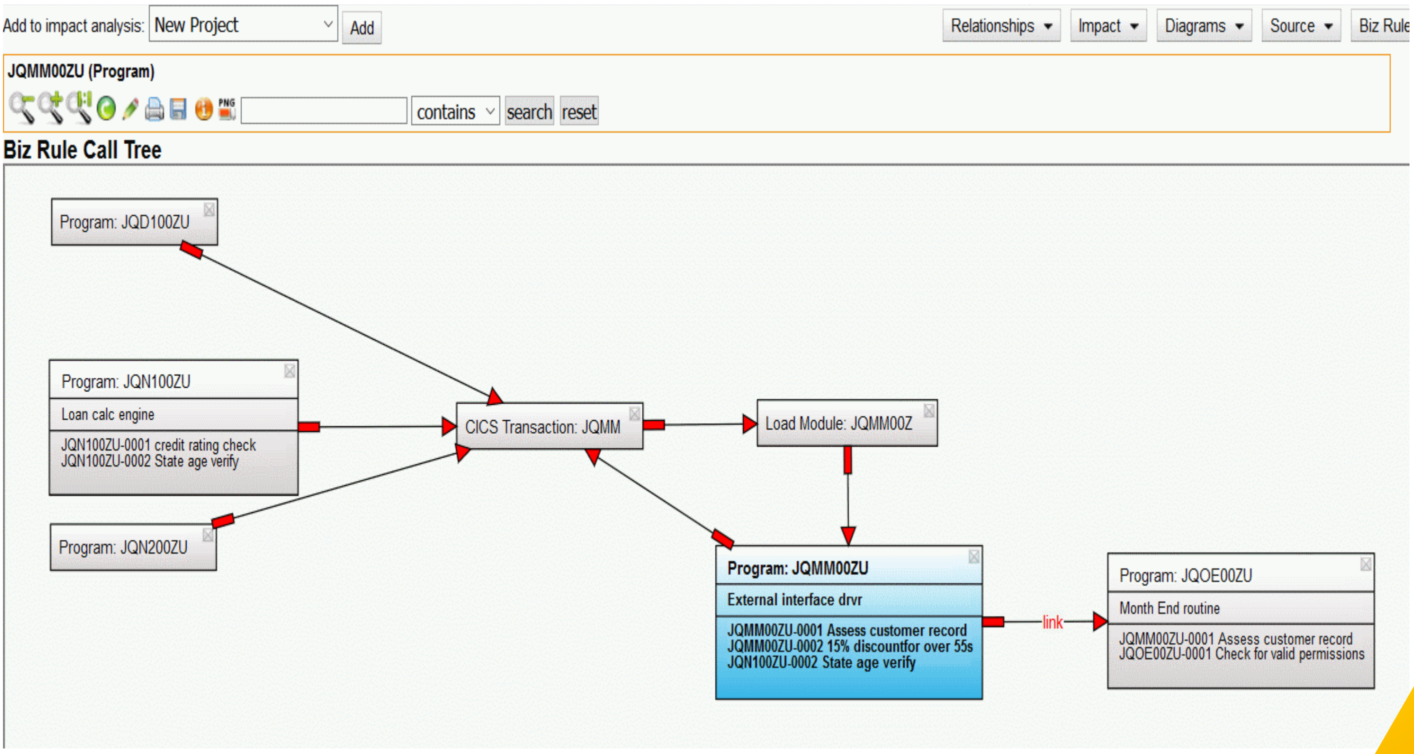
Supported technologies

COBOL, S-COBOL, T-COBOL, COBOL VME, Micro Focus ACUCOBOL-GT, Micro Focus COBOL, HP COBOL

About Advanced

Advanced is a leading provider of innovative and pioneering modernisation solutions, with thousands of organisations worldwide using our products and services. Our aim is to enable our customers to increase business value and maintain competitive advantage through maximising the potential of existing data and applications. This provides a rapid return on investment, a reduction in costs, improved productivity and efficiency with the ability to manage operational risk.

With over 30 years of experience, and expert staff who are dedicated to servicing the needs of organisations with legacy systems, we pride ourselves in our tailored approach to organisations. Major organisations including Deutsche Bank, Citi, Gap, International Paper, Chevron, Associated Wholesale Grocers, Genunie Parts Co., The Home Depot and many other well-known firms have enjoyed the business benefits of an Advanced approach to legacy applications.



More information

w oneadvanced.com
t +44(0) 8451 605 555
e hello@oneadvanced.com

Ditton Park, Riding Court Road, Datchet, SL3 9LL

Advanced Computer Software Group Limited is a company registered in England and Wales under company number 05965280, whose registered office is Ditton Park, Riding Court Road, Datchet, SL3 9LL. A full list of its trading subsidiaries is available at www.oneadvanced.com/legal-privacy.