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 organization. Business Rules represent the intellectual property of the organization, provdiing competitive differentiation and business optimization.

Business Rules are deeply embedded in the existing COBOL program's and depending on how 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 organizations, 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 organizations 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.

Business Rules Extraction (BRX) with Application Analyser

Business Rules hidden in an application is to combine the subject matter expertize 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 semanitc 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 cadidate data 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 indetified and documents, 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

show/hide toc | dashboard | search: [_____] ▾ | name ▾ | go | advanced search | search history | impact analysis | print | settings | user guide

Relationships ▾ Impact ▾ Diagrams ▾ Source ▾ Biz Rules ▾

JQMM00ZU (Program)

Program Description: External interface drvr

all ▾ Stmt Type	Line #	Weighting	WS	FD	LS	CB	DCLGEN	all ▾ Business Rule? (Y/N/M)	Rule Id
			Variable Types Used						
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		▾	

```
1131  MOVE WS-USERID        TO JRNL-DB-USERID.
1131  MOVE EIBTRMID         TO JRNL-DB-TERMID.
1132  MOVE EIBRESP          TO JRNL-DB-EIBRESP.
1133  MOVE CM-SELECT-OPTION TO JRNL-DB-SELECT.
1134  MOVE CM-SEARCH-OPTION TO JRNL-DB-SEARCH.
1135  MOVE CM-SEARCH-VALUE  TO JRNL-DB-VALUE.
1136  MOVE   SQLCODE         TO WS-SYS-ERRCODE.
1137  CALL 'DSNTIAR'   USING SQLCA
1138                         JRNL-DB-ERROR-MSG
1139                         ERROR-MSG-LEN.
1140  MOVE ERROR-LINE-DEF   TO JRNL-DB-MSG.
1141  EXEC CICS SYNCPOINT ROLLBACK
1142              RESP(RESP-CODE)
1143              END-EXEC.
1144  EXEC CICS ASKTIME ABSTIME(WS-CABSTIME)
1145                    RESP(RESP-CODE)
1146              END-EXEC.
1147  EXEC CICS FORMATTIME ABSTIME(WS-CABSTIME)
1148              TIME(JRNL-DB-TIME)
1149              RESP(RESP-CODE)
1150          END-EXEC.
1151  EXEC CICS WRITE FILE('JQLOG')
1152              FROM(JRNL-DB-LOG)
1153              LENGTH(JRNL-DB-LEN)
1154              RIDFLD(JRNL-ER-KEY)
1155              RBA
1156              RESP(RESP-CODE)
1157          END-EXEC.
1158  MOVE   'SQL ERROR, CONTACT DEVELOPERS' TO WS-SYS-ERRM
1159  EXEC CICS SEND  FROM (SYSTEM-MESSAGE)
1160                  LENGTH(70)
1161                  ERASE
1162                  RESP(RESP-CODE)
1163              END-EXEC.
1164          IF RESP-CODE NOT = NORMAL
1165              MOVE RESP-CODE TO  RTNN
1166              GO TO Z9999-CICS-ERROR
1167          END-IF.
1168      MOVE 1 TO CM-JQOE-RETFLAG.
1169      EXEC CICS RETURN END-EXEC.
1170      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

> Users are configured to your business and application environments

> A wide variety of COBOL compilers are supported

> Business Rules are documented in English

> Display Business Rules in Process Flows, referencing underlying programs

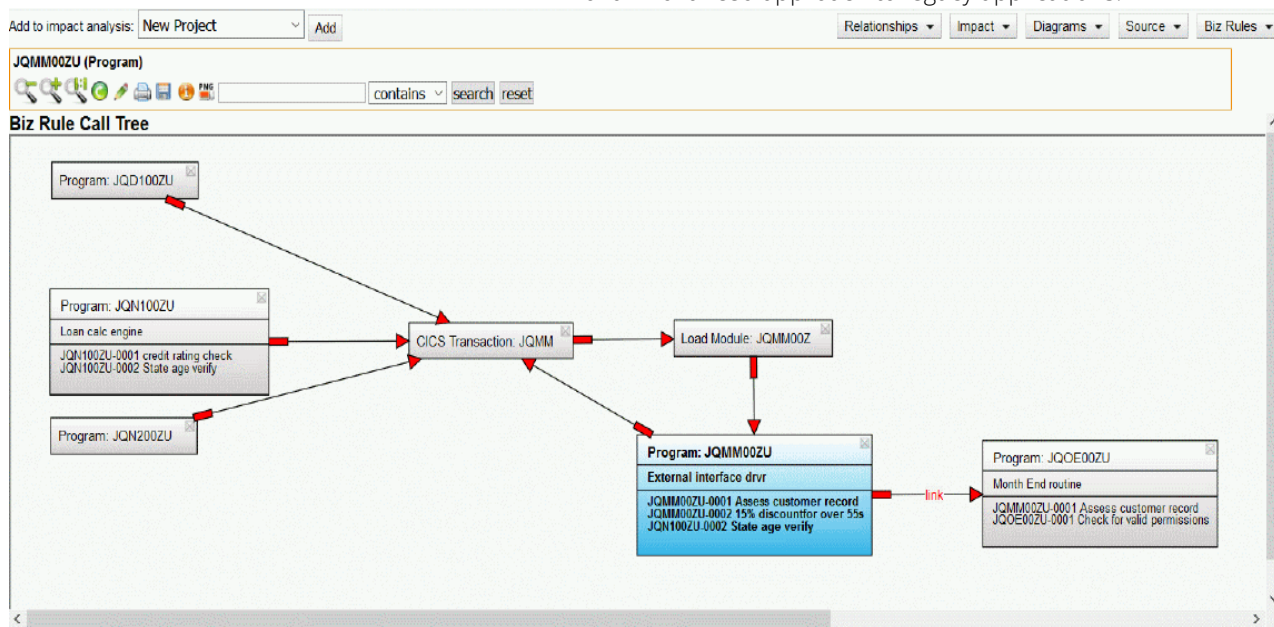> BRX builds on the proven capabilities of our Application Analyser solution

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 modernization solutions, with thousands of organizations 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 organizations with legacy systems, we pride ourselves in our tailored approach to organizations. Major organizations 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 770 933 1965
e hi@oneadvanced.com

3200 Windy Hill Road, Suite 230 West, Atlanta, GA 30339