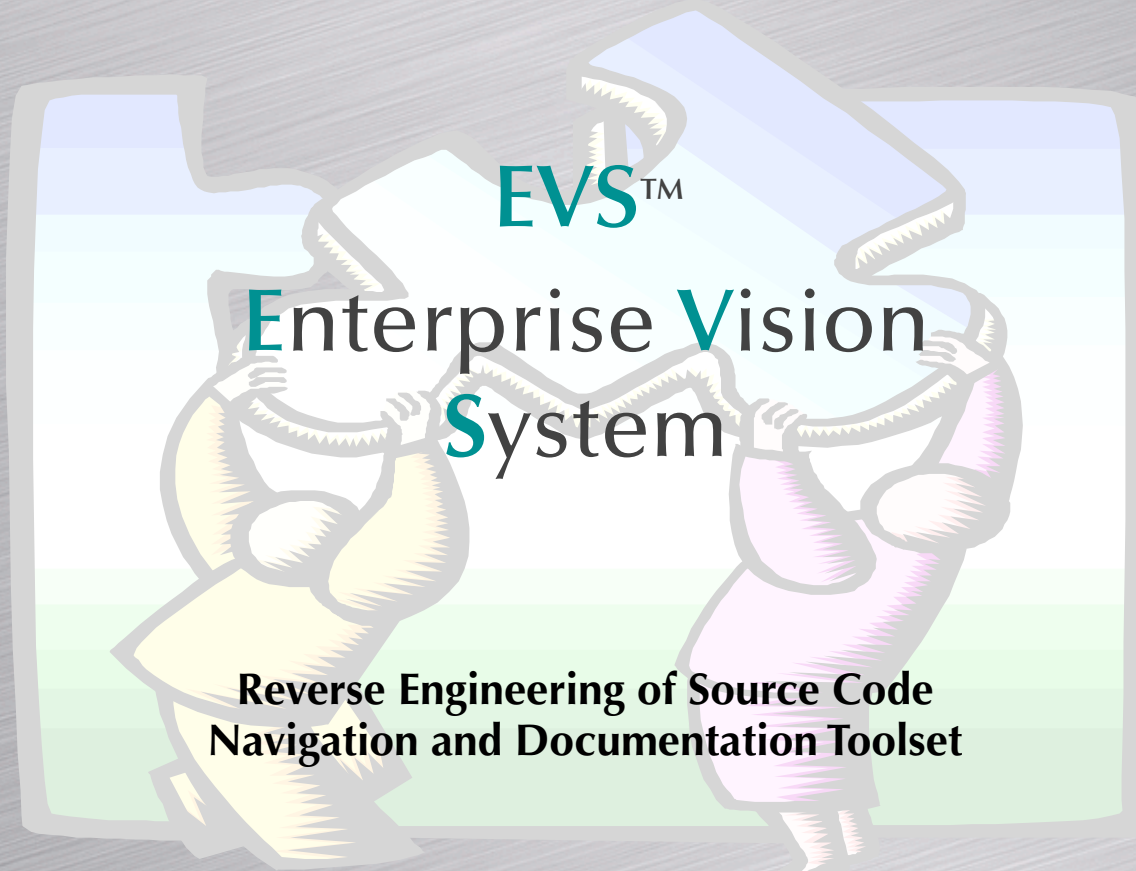


Xactis



**Reverse Engineering of Source Code
Navigation and Documentation Toolset**

An **Xactis** Technical Overview

EVS™ is a proprietary toolset that extracts all meaning from legacy source code and provides a navigation mechanism to simplify the migration, maintenance, modification, testing and quality assurance of legacy systems.

Exactis code analysis toolsets move the industry towards the automation of software migration, maintenance, testing and QA.

EVS – A New Static Analysis Technology

The Exactis code documentation and navigation technology successfully accomplishes the following:

- ***Transforms*** source code into a relational semantic database that classifies all syntactic elements, including business rules and variable flows – inside and between every program and system
- ***Catalogs*** all statements, procedures, parameters, data definitions, references, including user and file I/O's and temporary variables
- ***Tracks*** all associations – everything identifiable that is contained in the code
- ***Delivers a rich feature-set***
 - Powerful and highly adaptable enterprise source code documentation tool
 - Fully searchable database for software engineering requirements
 - Fast, interactive user interface to find what is needed
 - Highly adaptable toolset - from the project level to the enterprise
 - Compatible with most programming languages

How EVS™ Extracts and Displays Source Code Components and Logic

**Cobol Source Code
and reference manuals are
submitted by the customer,
then code is 'quantized.'**

A new highly-adaptive compiler technology developed by Exactis, to extract syntactic and semantic information from all varieties of COBOL source code.

How EVS™ Extracts and Displays Source Code Components and Logic

**Cobol Source Code
and reference manuals are
submitted by the customer,
then code is 'quantized.'**

**Language Module (LM)
Developed per exact COBOL
dialect with programming nuances
taken into account.**

Any variations off of ANSI standards are easily accommodated before "quantization".

How EVS™ Extracts and Displays Source Code Components and Logic

**Cobol Source Code
and reference manuals are
submitted by the customer,
then code is 'quantized.'**

**Language Module (LM)
Developed per exact COBOL
dialect with programming nuances
taken into account.**

**Additional Source Code &
Reference Manuals Submitted
Custom language modules are built,
then code is 'quantized.'**

Additional languages will be integrated on a custom basis.

How EVS™ Extracts and Displays Source Code Components and Logic

**Cobol Source Code
and reference manuals are
submitted by the customer,
then code is 'quantized.'**



**Additional Source Code &
Reference Manuals Submitted**
Custom language modules are built,
then code is 'quantized.'

Language Module (LM)
Developed per exact COBOL
dialect with programming nuances
taken into account.



Additional Language Module(s)
To handle source files, libraries,
subroutines, stored procedures, etc.
in all other programming languages.

How EVS™ Extracts and Displays Source Code Components and Logic

Cobol Source Code
and reference manuals are
submitted by the customer,
then code is 'quantized.'



**Additional Source Code &
Reference Manuals Submitted**
Custom language modules are built,
then code is 'quantized.'



Knowledge Base
of code elements & logic
generated to populate this
highly searchable database.

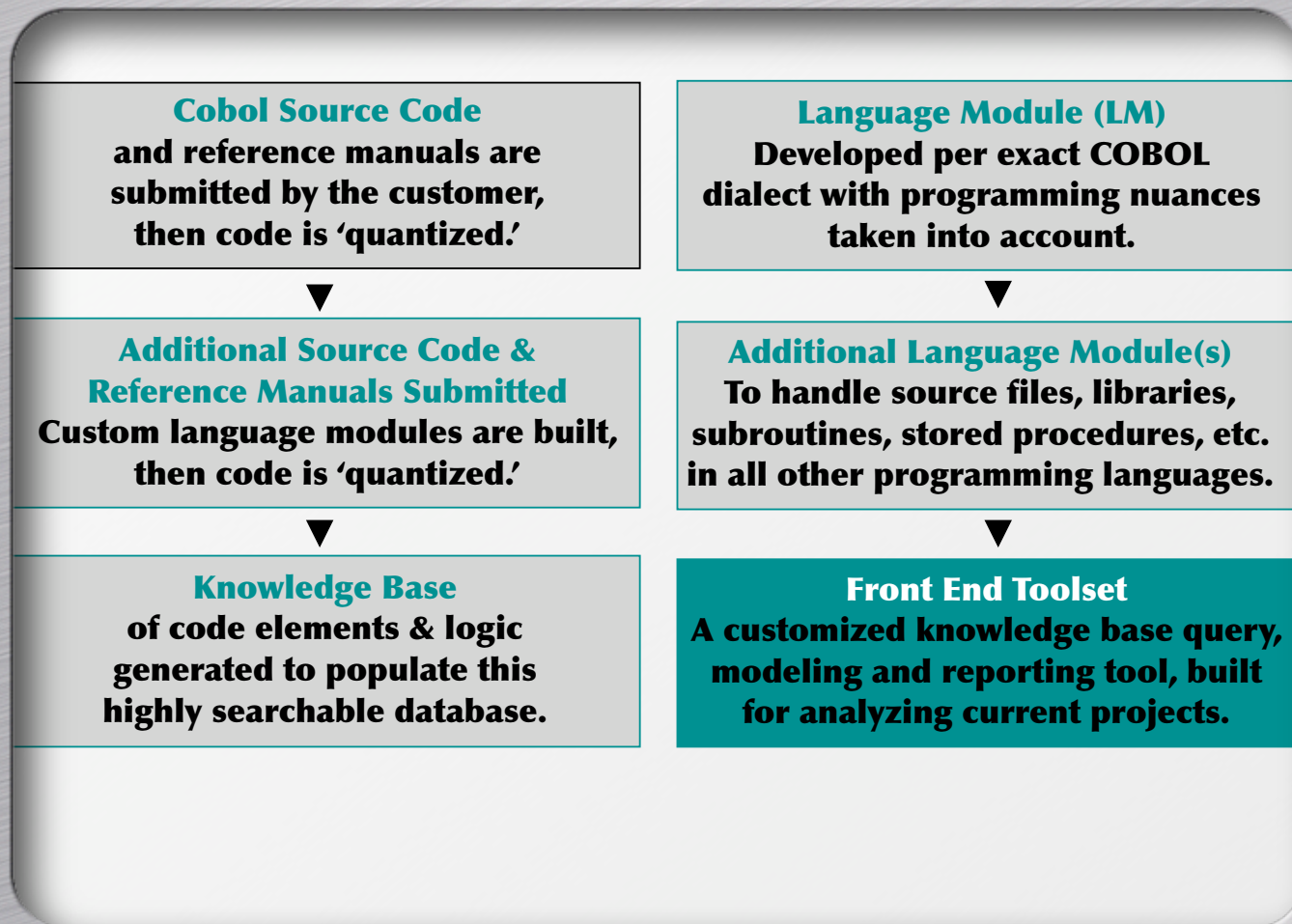
Language Module (LM)
Developed per exact COBOL
dialect with programming nuances
taken into account.



Additional Language Module(s)
To handle source files, libraries,
subroutines, stored procedures, etc.
in all other programming languages.

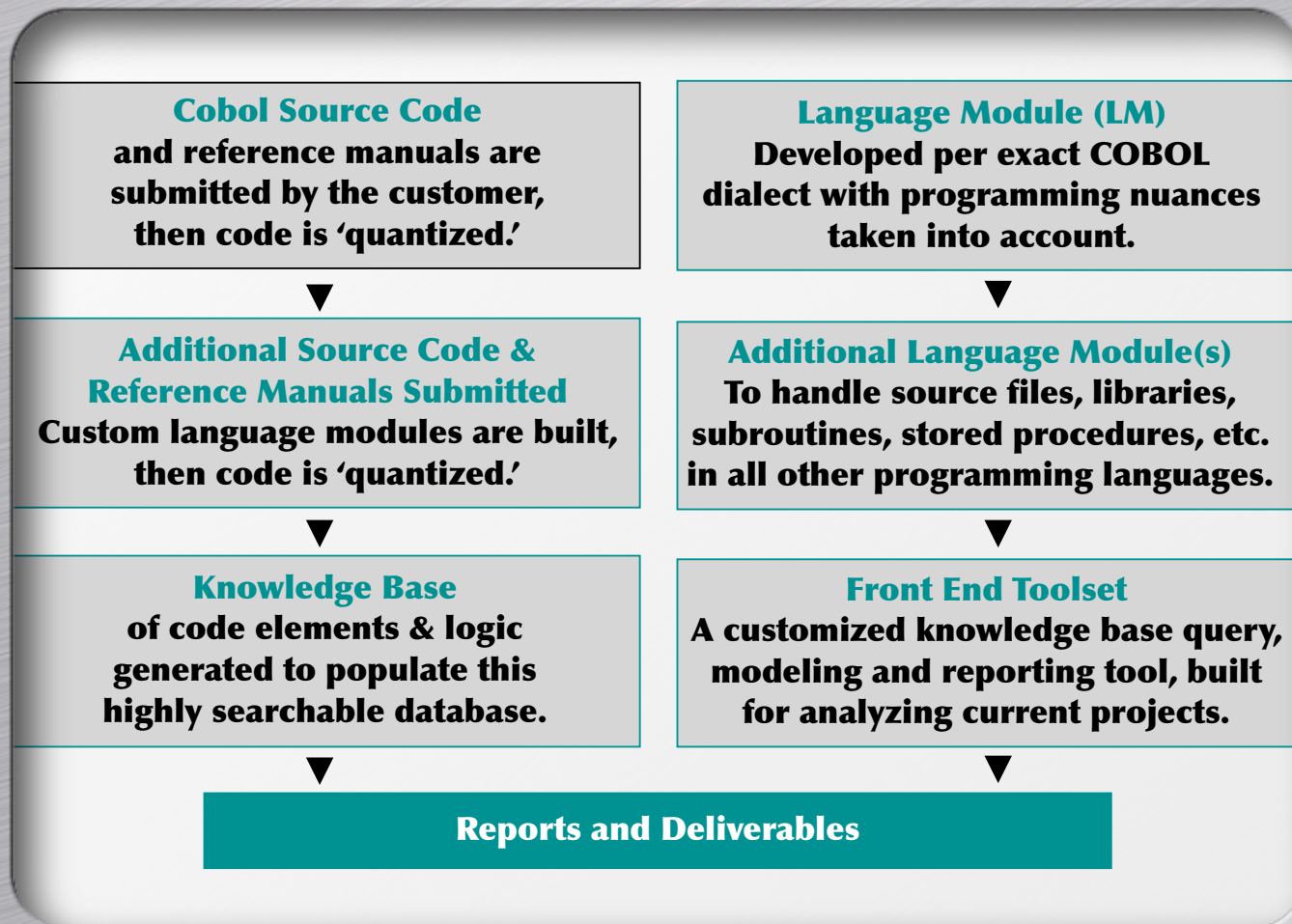
The quantized *qlogiq*™ file that contains all source code information is transformed into a relational database.

How EVS™ Extracts and Displays Source Code Components and Logic



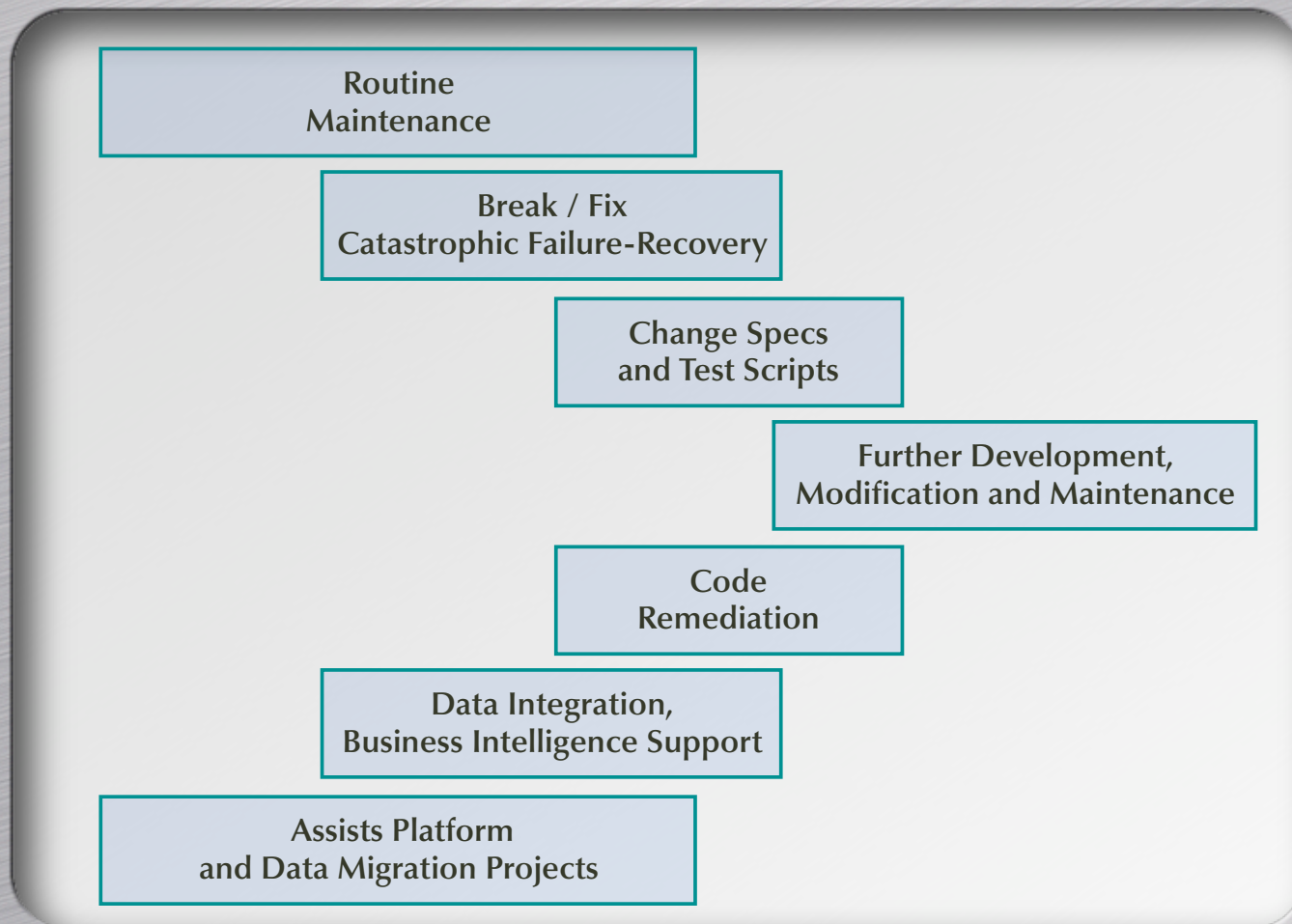
The EVS can be modified, enhancing productivity as project requirements change.

How EVS™ Extracts and Displays Source Code Components and Logic



Custom reports are generated by EVS,
in addition to helping the programmer navigate the code.

Uses for the EVS



Quickly resolves anomalies that plague migration and integration efforts by knowing all fields and their relationships

Statistics Manager

Statistics Manager
_ □ ×

Full Information

Cobol Files (*.cob) Total: 5

- APPBSPYM.cbl
- GLPJEHDR.cbl
- glpoelim.cbl
- PTPDYSQL.cbl
- PTPRECUR.cbl

Copy Files (*.cpy) Total: 58

- APCBCNTL.cpy
- APCBBUSD.cpy
- APCBFCLS.cpy
- APCBBMSG.cpy
- APCBICLS.cpy
- APCBISTM.cpy
- APCBORGD.cpy
- APCBOPTB.....

Count of Listed Fields: 354

- ANSWER
- BANK-ACCT-KEY
- BANK-ACCT-KEY
- BANK-ACCT-KEY-S
- BANK-CD
- BANK-CD
- BANK-CD-S
- BANK-CD-TD

Specific Information

APPBSPYM.cbl

Total Lines	3076
Count of Declared Variables	354
Count of associations	0

Project Summary

Total Lines	15640 in 63 files.
Total associated fields in all files	843
Number of relations	3885

Selected Variable Details

Count of Associations	0
Count of Relations	0

Number of occurrences

Statement Type	If Condition	247
Data Field		0

File Name	First Line of code in statement
GLCJLOGM.cpy	201\..... IF DYSQL-LOG-YES OF SQLRT
GLPJEHDR.cbl	203\..... IF DYSQL-LOG-OPEN-EXTEND OF SQLRT
GLPOELIM.cbl	211\..... IF FILE-STAT-OK OF W-WK
PTCDBDTM.cpy	213\..... IF NOT DYSQL-LOG-OPEN-EXTEND OF SQLRT
PTCSYSD.cpy	247\..... IF DYSQL-LOG-YES OF SQLRT
PTPDYSQL.cbl	259\..... IF DYSQL-LOG-YES OF SQLRT
PTPRECUR.cbl	281\..... IF FIRST-CALL-YES OF DYSQL
	288\..... IF NOT RTNCD-OK OF SQLRT
	334\..... IF WORKDISP OF W-WK + 72
	361\..... IF NOT FILE-STAT-OK OF W-WK
	380\..... IF COMMON-CURSOR-YES OF DYSQL
	415\..... IF COMMON-CURSOR-YES OF DYSQL

[View Entire File](#)

Previous Next

The EVS™ Statistics Manager offers a high-level view of application file contents and metrics - for the management of code maintaining and changing efforts, to allocate project funds and human resources.

Variable Definitions and Relationships

The EVS toolset shows . . .

- ALL data fields, definitions, families and occurrences
- Variable relationships in CONTEXT - which fields impact and are impacted by others
- Joint occurrences of multiple data fields and the actual line of source code where the variable is referenced
- All statements that use the variable including IF conditions, calls, moves, etc.

Fields

ACCOUNT-DESCR-ROUND
ACCOUNT-DESCR-SUSPENSE
ACCOUNT-FETCHED-NO
ACCOUNT-LEN
ACCOUNT-ROUND-ADJ

Total: 843

Field Attributes

Index	993
Level	2
File Name	GLPJEHDR.cbl
Line #	48
Pic	X(254)
Parent	996

Relationships

If Condition
Move Statement
Call Statement
Move Statement

Total: 5

Relationship Attributes

Index	638
Group	226
File Name	GLPJEHDR.cbl
Line #	968

Related Fields

ACCOUNT-DESCR-ROUND
ACCOUNT-DESCR

Total: 2

Related Field Attributes

Index	993
Group	638
Type	Move From
Pic	X(254)
Line #	48
In File	GLPJEHDR.cbl

Field Family W-W...
1 W-WRK

Entry Point for: ACCOUNT-DESCR-ROUND
ACCOUNT-DESCR-ROUND.DEV/W-WRK

File Listing For: GLPJEHDR.cbl

```

961 ..... IF ACCOUNT-DESCR-ROUND OF W-WRK = SPACES
962 .....
963 .....     PERFORM H1110-GET-ACCOUNT-DESCR-ROUND
964 .....     END-IF
965 ..... END-IF
966 .....
967 ..... IF STATUS-OK OF JEDIT
968 .....     MOVE ACCOUNT-DESCR-ROUND OF W-WRK TO
969 .....         ACCOUNT-DESCR OF JRNL
970 .....     MOVE JOURNAL-ID OF JEDIT TO REFERENCE-VALUE OF JRNL
971 .....     MOVE SPACE TO CUR-RT-TYPE OF JRNL
972 .....     MOVE ZERO TO CUR-EXCHNG-RT OF JRNL
973 .....     COMPUTE MON-AMT OF JRNL = JRNL-TOT-CREDITS OF
974 .....         BALANCES-WRK OF JEBAL - JRNL-TOTAL-DEBITS OF
975 .....         BALANCES-WRK OF JEBAL
976 .....     MOVE ZERO TO FRN-AMT OF JRNL
977 .....
  
```

Path for: ACCOUNT-DESCR-ROUND

```

ACCOUNT-DESCR-ROUND
├── ACCOUNT-DESCR
│   ├── SETUP-DATA-PTR
│   │   ├── ACCOUNT-DESCR-SUSPENSE
│   │   │   ├── ACCOUNT-DESCR
│   │   │   └── SUSPENSE-ACCOUNT-ERROR
│   └── MAX-LINE-AMOUNT
│       ├── JRNL-TOTAL-DEBITS
│       │   ├── BASE-TOTAL-DEBITS
│       │   ├── BASE-TOT-CREDITS
│       │   ├── UNBALANCED-NO
│       │   ├── CURRENCY-CD
│       │   ├── FOREIGN-CURRENCY
│       │   └── JRNL-CNTL-DEBITS
│           └── SETUP-DATA-PTR
│               ├── BUSINESS-UNIT
│               └── BALANCE-OF-VALUE
  
```

Rules Embedded in Conditional Logic

The screenshot displays a software development environment with several components:

- Fields Panel:** Lists fields like BASE-TOTAL-DEBITS, BEGIN-DT, and relationships like Unknown Statement, Move Statement, If Condition, and Compute Statement.
- Main Code Editor:** Shows source code lines 1409 through 1430. Line 1409 is highlighted: `IF JRNL-TOTAL-DEBITS OF BALANCES-WRK OF JEBAL`. Other lines include `NOT = BASE-TOTAL-DEBITS OF W-WRK`, `OR JRNL-TOT-CREDITS OF BALANCES-WRK OF JEBAL`, `NOT = BASE-TOT-CREDITS OF W-WRK`, `SET BAL-FIELDS-MATCH-NO OF W-FLG TO TRUE`, `PERFORM VARYING BAL2-IDX FROM 1 BY 1`, `UNTIL BAL2-IDX > BALANCES-COUNT`, `OR BAL-FIELDS-MATCH-YES OF W-FLG`, `IF BUSINESS-UNIT OF BALANCES-WRK OF JEBAL = BUSINESS-UNIT OF BALANCES OF JEBAL (BAL2-IDX)`, `AND CURRENCY-CD OF BALANCES-WRK OF JEBAL = CURRENCY-CD OF BALANCES OF JEBAL (BAL2-IDX)`, `AND FOREIGN-CURRENCY OF BALANCES-WRK OF JEBAL = FOREIGN-CURRENCY OF BALANCES OF JEBAL (BAL2-IDX)`, and `SET BAL-FIELDS-MATCH-YES OF W-FLG TO TRUE`.
- If Condition Logic Tree:** A hierarchical diagram showing the nested structure of the code's conditional logic, including 'Set Statement', 'Perform Statement', and 'If Condition' blocks.
- File Listing For: GLPJHDR.cbl:** A window showing the extracted code as a list of lines with line numbers and asterisks, matching the main code editor.

The EVS Rule Extraction Process . . .
Extracts Business Rules from IF conditions
Shows all related statements
Displays BOTH code and logic and
Yields a more complete understanding
of the business knowledge embedded
in the source code

EVS Management Features and Benefits

- **Lowers the cost, risk, and duration of IT projects**
- **Rescues failing projects that lack legacy understanding and technical documentation of the systems**
- **Cuts maintenance budgets, testing, and break / fix efforts – often by more than half**
- **Automated documentation assists IT architecture design, development and integration interoperability**
- **Effective QA - detects errors of omission and commission**
- **Eliminates guesswork and facilitates 100% testing of only those paths that need to be checked**
- **Drastically reduces the cost of the entire testing process**
- **Provides data mart/data warehouse assistance to resolve anomalies and maximize decision support information**
- **Improves the morale and productivity of programmers**

EVS Technical Features and Benefits

Software Maintenance, Modification, Upgrading, and Testing

- **Track Code** while modifying every effected variable of every changed variable throughout all applications - no more patching
- **Ensure** complete test case coverage for all events, processes, rules, and data relationships created in the source code - determining exactly where and what to test
- **Use** on all types of code in multi-language environments: COBOL, C, C++, SQL, Java, etc.

System / Platform Migration as a Maintenance Activity

- **Extract** business rules embedded in source code
- **Show** source code and variables related to every business rule
- **Ensure** inclusion of all business rules
- **Create** documentation to include “spoken business rules”
- **Duplicate** legacy functionality in new languages

Data Validation and Business Rules Verification for Data Quality, Data Integration, and Data Mart / Data Warehousing Efforts

- **Ensure** data elements from databases are used consistently across business processes
- **Analyze** subtle, everyday problems, without programming experts - finding and resolving homonyms, synonyms, hidden fields, version differences
- **Obtain** a deeper level of focus on business rules - how applications use and represent data (especially useful when integrating variable length flat files)

Worldwide Impact of EVS

US Department of Education

- Helped determine the feasibility of combining 12 student loan systems into one - saving congress 2.2 billion dollars in appropriations
- Cited in the United States Congressional Record as a meaningful technology to track “inter-operability” between systems and agencies
- Cited in the Congressional Record by the GAO, who called the tool an “analysis technology able to check for systems interoperability.”

US Department of Defense

- Produced automated Documentation and Rule Extraction for system controlling spare parts ordering

Texas Department of Human Services

- Helped complete a Y2K IV&V and the largest remediation effort for the largest mainframe in the state - The Welfare System
- UNiSYS mainframe, clean-up of obsolete / unused program code and files

Republic of Ecuador / Government and Corporate Activities

- Handled critical systems work at the telephone company (ANDINATEL)
- Oil monopoly (PetroEcuador)
- Social Security Agency (IESS)

AT&T, FDIC and others