



AUTOMATED REFACTORING FROM LEGACY CODE TO MAINTAINABLE JAVA

CM evolveIT Meta TX-AI can refactor your legacy app developed in native COBOL, or CA 2E (Synon) generating RPG or COBOL directly to maintainable runtime-free Java and JS.

With the help of CM evolveIT Meta TX-AI's automated code refactoring superpower, you can effectively convert your legacy codebases to runtime-free Java and JS. Your applications will operate on the cutting edge, and future developers will be able to get up to speed quickly.

Understanding automated code refactoring

Code refactoring is a disciplined technique aimed at improving the internal structure of software code without altering its external functionality. It involves making incremental changes to the codebase to enhance readability, reduce complexity, and eliminate redundancy.

Benefits of refactoring to Java

- + You will have a **greater resource pool** of Java and JS programmers.
- + You will be **cloud-ready** for all major cloud platforms.
- + You can **reduce computing costs**. Employing the IBM Z zIIP processor is just one example.
- + Refactoring **reduces technical debt** by eliminating design flaws, duplicated code, and obsolete constructs, leading to better long-term productivity.
- + **Easier to integrate Java** with external processes like AI.

CM evolveIT Meta TX-AI provides several features that facilitate refactoring legacy code to Java and JS

Code Analysis

CM evolveIT Meta TX-AI performs an in-depth analysis of the legacy codebase, identifying potential areas for improvement. It examines program structure, data usage, and logic flow, enabling developers to identify opportunities for refactoring.

Code Metrics

The tool generates comprehensive reports on code quality, complexity, and maintainability. These metrics help developers prioritize refactoring efforts and track the progress of code improvements.

Automated Refactoring

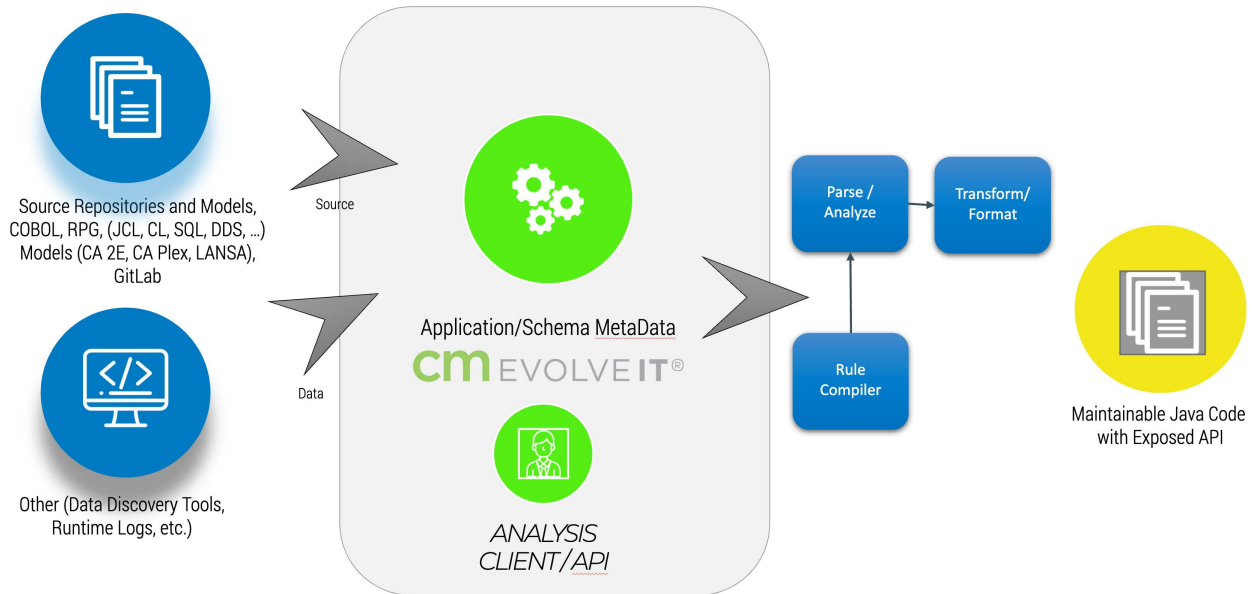
CM evolveIT offers a range of automated refactoring techniques specific to COBOL. From restructuring data structures to improving naming conventions and eliminating dead code, these refactoring actions are applied programmatically, saving valuable development time.

Dependency Visualization

The tool visualizes code dependencies, allowing developers to identify and refactor complex interdependencies. This feature helps to improve code modularity and reduce tight coupling.

OUR APPROACH

Refactoring an application developed in a legacy language to maintainable runtime-free Java requires strategy and planning. We stress a strong upfront analysis and careful planning in designing the right technology stack with CM evolveIT Meta TX-AI at the core, avoiding unnecessary vendor lock-in.



The Road to Cost Savings

The cost-savings benefits of a COBOL to Java conversion can't be stressed enough. You'll graduate to a more efficient and powerful modernized application and spend less money on the labor required to run it.

+ Bigger Labor Pool

COBOL programmers are in dwindling supply. Conversely, Java is the go-to choice for newly-trained programmers, creating a labor pool that allows you to staff up with coders who can grow with your application, and find replacements relatively easily.

+ More Efficient Hardware

The move to Java will also land you in a more efficient hardware ecosystem. You can leverage things like specialty processors to offload workloads running on general processors with metered licensing fees, saving you big money.

+ Cleaner Architecture

And let's not forget about the software itself. Running on a modern language like Java means more efficient use of all associated resources, saving you big money on processing, storage, and maintenance.

THE POWER OF AUTOMATION

When refactoring your codebase, automation is a powerful ally loaded with high-impact benefits, decreasing costs, time, and risk in one concerted effort. When tuned correctly CM evolveIT Meta TX-AI automation will not only minimize end-user training allowing you to utilize your existing staff, it's highly automated at scale and maintainable within standard architectures. You'll be able to plug it into your standard DevOps pipelines and go.

MSDHPVR ENTER PMT DMS POC

Please Enter Product ID and QTY

Product ID 11111 QTY 50

Product Name **Orange 100%**
Product Price **5.00**
Discount Rate **.10**
Discount Threshold **100.00**

Total Proce **225.00**
Discount Applied? **Y**

cmEVOLVE IT® Home Order Calculation

Product ID QTY

Order Result	
Product Name	Orange 100%
Product Price	\$5
Discount Rate	10%
Discount Threshold	\$100
Total Price	\$225
Discount Applied	Y



```
Get Product info
DMS Get Product EXT - DMS POC PRODUCT *
SET C-INDICATOR-OFF(90) TO TRUE
MOVE PIABNB TO WQ0001
MOVE SPACES TO WQ0002
MOVE ZEROS TO WQ0003
MOVE ZEROS TO WQ0004
MOVE ZEROS TO WQ0005

CALL 'MSDIXFR' USING
Return code
W0RTN
PRD Product ID
WQ0001
PRD Product Name
WQ0002
PRD Product Price
WQ0003
PRD Product Discount
WQ0004
PRD Discount Threshold
WQ0005
ON OVERFLOW
SET C-INDICATOR-ON(90) TO TRUE
END-CALL
```

```
// * get product info
// * dms get product ext - dms poc product *
indics.getIndImpl()[89].setIndicatorOff();
setWq0001(plparm.getPlabnb());
setWq0002("");
setWq0003(java.math.BigDecimal.valueOf(0));
setWq0004(java.math.BigDecimal.valueOf(0));
setWq0005(java.math.BigDecimal.valueOf(0));
// *
if ( !Code.tryCall("MSDIXFR", w0rtnImpl.byReference(), wq0001Impl.b
indics.getIndImpl()[89].setIndicatorOn();
}
// *
workContext.y10003 = getWq0002();
workContext.y10004 = getWq0003();
workContext.y10005 = getWq0004();
workContext.y10006 = getWq0005();
```

+ Before Automation

Your green-screen application is trusty and reliable, but lacks the modern features and flexibility you need to compete in today's competitive technical marketplace.

+ After Automation

Your application is refactored into maintainable runtime-free Java accessible by the web, maintainable by a much larger labor pool, and no vendor lock-in is required.

NOW WITH CHATGPT INTEGRATION

Generative AI is throwing the software world for a loop with dramatic productivity gains. The promise is real if used intelligently and with skill. The key: inject AI access where the important work is done and focus its power within a carefully selected universe of context-aware prompts.

+ AI Output

Artificial Intelligence outputs explains and analyzes your existing code, and generates new code as needed.

The screenshot displays the MSDDXFR application interface. At the top, a header bar shows the application name 'MSDDXFR' and its type 'applicationprogram' with the description 'Calculate Order'. Below this, a 'Source Code View' section contains a dropdown menu set to 'All' and a 'Request AI Analysis' button. The main area is divided into two panes: 'Original' on the left and 'Result' on the right. The 'Original' pane shows COBOL source code for calculating a product price, including variables like 'PRD Total Price', 'LCL-PRD Discount Threshold', and 'LCL-PRD Product Discount'. The 'Result' pane shows the AI's explanation of the code, detailing the logic for calculating the total price and applying a discount. A context menu is open over the 'Result' pane, listing various actions such as 'Convert DDS to SQL DDL with friendly names', 'Explain Source', 'Convert COBOL Linkage Section to Rest API Request Model', 'Simplify Nested IF block', 'Convert Nested IF to Evaluate', 'Refactoring Code Suggestion', 'Convert Data Structure to SQL DDL', 'Convert DDS to SQL DDL', 'Convert DDS to SQL DDL with friendly names', 'Convert SQL DDL to SQL CURD for DB2/400', and 'Convert SQL DDL to SQL CURD For'.

Original

```
1 * Calculate
2 * PRD Total Price
3 * MULTIPLY YL0004 OF WORK-CONTEXT BY P2ARNB GIVING YL0002 OF
4 * WORK-CONTEXT
5 * CASE: LCL-PRD Total Price GE LCL-PRD Discount Threshold
6 * *IF
7 * IF (YL0002 OF WORK-CONTEXT NOT < YL0006 OF WORK-CONTEXT) THEN
8 * LCL-PRD Total Price =
9 * COMPUTE (X1 * (X2 - X3))
10 * X1 : LCL-PRD Total Price
11 * * : PGM.*Synon (17,7) work field *
12 * X2 : CON.1
13 * * : PGM.*Synon (17,7) work field
14 * X3 : LCL-PRD Product Discount
15 * *Synon (17,7) work field
16 * SUBTRACT YL0005 OF WORK-CONTEXT FROM 1 GIVING W00001
17 * *Synon (17,7) work field
18 * MULTIPLY YL0002 OF WORK-CONTEXT BY W00001 GIVING W00000
19 * PRD Total Price
20 * MOVE W00000 TO YL0002 OF WORK-CONTEXT
21 * Product Discount Applied
22 * MOVE 'Y' TO YL0001 OF WORK-CONTEXT
23 * END-IF
```

Result

```
1 This COBOL code is used to calculate the total price of a product (PRD Total Price).
2
3 The code first multiplies the value of YL0004 OF WORK-CONTEXT by P2ARNB and stores the result in YL0002 OF WORK-CONTEXT.
4
5 Next, the code checks if the PRD Total Price is greater than or equal to a discount threshold (YL0006 OF WORK-CONTEXT). If it is not, the code will not perform any further calculations.
6
7 If the PRD Total Price is greater than or equal to the discount threshold, the code subtracts the value of YL0005 OF WORK-CONTEXT from 1 and stores the result in W00001. It then multiplies the value of YL0002 OF WORK-CONTEXT by W00001 and stores the result in W00000. Finally, the code moves the value of W00000 to YL0002 OF WORK-CONTEXT, indicating the updated PRD Total Price.
8
9 Additionally, the code sets the value of YL0001 OF WORK-CONTEXT to 'Y' to indicate that a product discount has been applied.
10
11 Overall, this code calculates the PRD Total Price of a product and applies a discount if the Total Price meets certain criteria.
```

Convert DDS to SQL DDL with friendly names

Explain Source

Convert COBOL Linkage Section to Rest API Request Model.

Simplify Nested IF block

Convert Nested IF to Evaluate

Refactoring Code Suggestion

Convert Data Structure to SQL DDL

Convert DDS to SQL DDL

Convert DDS to SQL DDL with friendly names

Convert SQL DDL to SQL CURD for DB2/400

Convert SQL DDL to SQL CURD For

+ Source Code Power

The CM evolveIT TXAI source code viewer provides a direct interface to the vast powers of AI, giving you all the problem-solving benefits of generative artificial intelligence at your fingertips.

+ Library of Prompts

Pre-defined prompts further add to the power by providing the exact queries needed to get the job done quickly and efficiently, focusing the user on the task at hand.

About CM First Products

CM First Group's powerful automation tools, augmented by services partners and professional services staff with many decades of software engineering and DevOps experience, ensure successful outcomes for even the most demanding modernization projects. Our products and expertise have helped over 400 customers in the public and private sectors reach their desired future state faster and more cost effectively than by using conventional approaches.

CM First software quickly analyzes, documents and re-platforms legacy code bases with minimal errors and rework, including those that are too large and complex for humans to tackle in any reasonable timeframe. The output is immediately usable by all team members, regardless of experience and knowledge of legacy software languages, accelerating application maintenance and modernization projects.

For more information, visit [**cmfirstgroup.com/products/CMevolveIT**](https://cmfirstgroup.com/products/CMevolveIT)

Request a Demo Today

Contact us for more information or to schedule a demo. Call 888-866-6179 or email us:

info@cmfirstgroup.com