COBOL Upgrade Advisor

Roland Koo

Program Director, Product Management and Strategy, Compilation Technologies, and Enterprise Products

rkoo@ca.ibm.com

Agenda

Staying current with COBOL

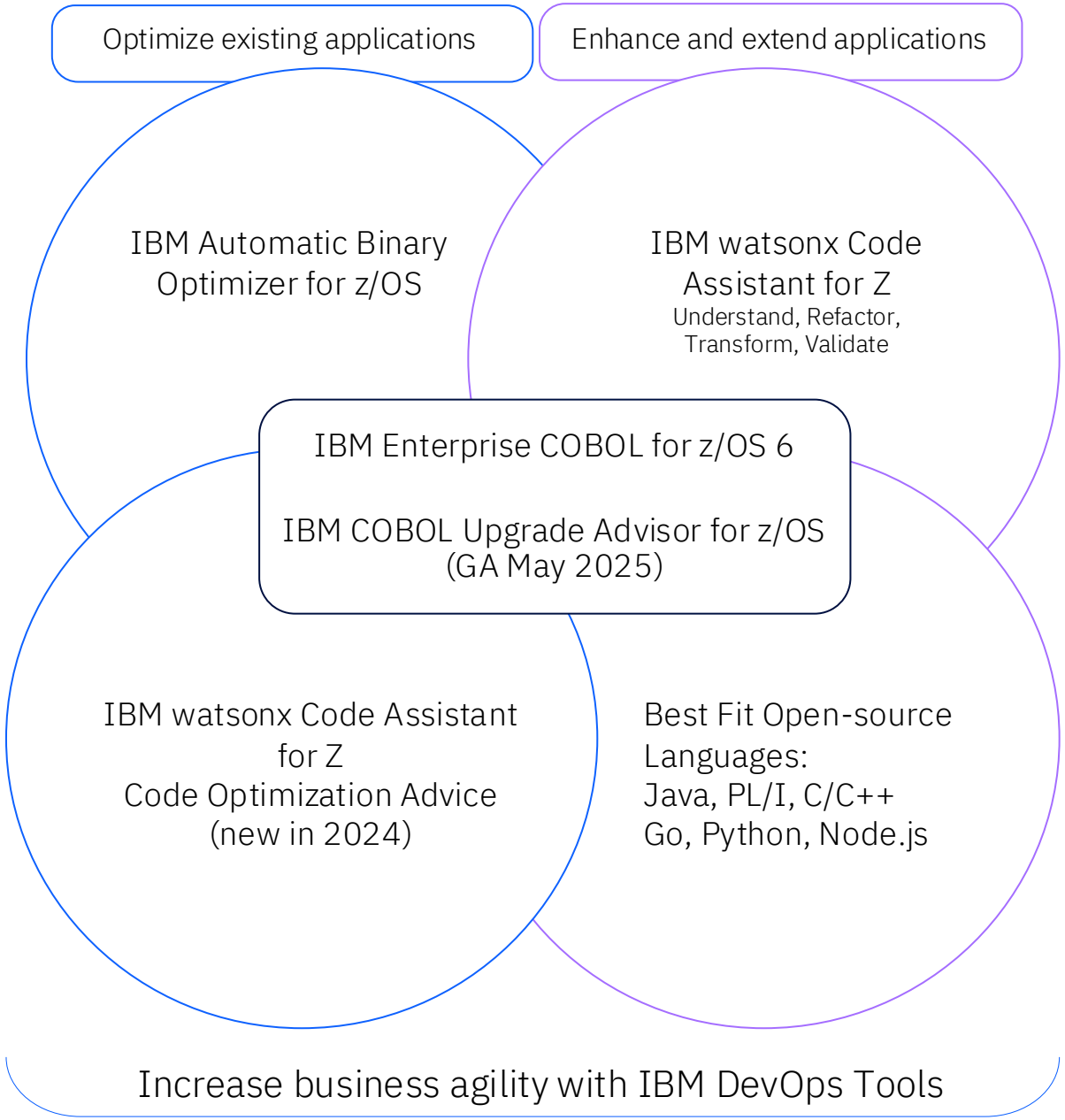
Introducing COBOL Upgrade Advisor (CUAZ)

CUAZ Demo

CUAZ Use Cases

Modernize COBOL Applications on z/OS

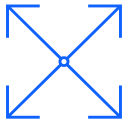
A unified approach to optimize, modernize, and extend COBOL workloads



Importance of upgrading to COBOL 6

Application performance is delivered by new hardware features and compiler support

Compile with COBOL 6 compiler to take full advantage of modern IBM z/Architecture and reduce operation costs



Get native support for modern technologies such as JSON, UTF-8, XML, AMODE 64 (64-bit), user-defined types and NoSQL database (VSAMDB) support

Get the most out of IBM watsonx Code Assistant for Z by upgrading to COBOL 6

Extend core applications with a new Java/COBOL interface

Detect and fix longstanding code and data issues in business-critical applications which are accumulated over years of development

Upgrade to a current IBM COBOL compiler to mitigate operational risks and ensure compliance with enterprise and regulatory policies that mandate the use of supported software

Application performance delivered by new IBM Z hardware features and compiler support

Decimal Floating Point zoned-conversion facility

Decimal-Floating-Point Packed-Conversion Facility, Vector Extension Facility

Vector packed-decimal facility

Vector packed-decimal enhancement facility

Vector packed-decimal enhancement facility 2

Vector packed-decimal enhancement facility 3

IBM zEC12/zBC12 ARCH=10	IBM z13/z13s ARCH=11	IBM z14/z14 ZR1 ARCH=12	IBM z15/z15 T02 ARCH=13	IBM z16 ARCH=14	IBM z17 ARCH=15
Improve performance of zone decimal arithmetic	Improve performance of packed decimal arithmetic and string operations	Key COBOL data types now first order hardware types	Multi-language application acceleration Faster conversions & computations	Exponentiation and float/packed performance Numeric-editing acceleration	Hardware accelerated data validations for faster COBOL upgrades Faster binary computations used for inter-lang apps
Enterprise COBOL 5.1 ABO 1.1	Enterprise COBOL 5.2 Enterprise COBOL 6.1 ABO 1.2	Enterprise COBOL 6.2 ABO 1.3	Enterprise COBOL 6.3 ABO 2.1	Enterprise COBOL 6.4 ABO 2.2	Enterprise COBOL 6.5 ABO 2.3

COBOL Enhancements (v5 and up)

COBOL 5.1 – 5.2

New advanced optimization framework (z10 – z13)

New COBOL runtime

New object and debugging format

Co-processor for SQLIMS

XML processing and UTF 8 enhancements

Language features/ enhancements: Table Sort, Exit, COPY, REPLACE, VOLATILE clause, Floating comment (*>)

COBOL 6.1

GA: 2016

Improved scalability with 64-bit optimization framework

Performance improvements (z13)

New language feature: ALLOCATE, FREE, INITIALIZE. **JSON GENERATE**

Improved reporting on invalid data and out of bounds for smoother upgrade from COBOL 4

COBOL 6.2

GA: 2017

Performance improvements (z14)

New language features: **Conditional compilation support, JSON PARSE**

New TEST sub-options: SEPARATE/NOSEPARATE

NOSTROPT option to suppress optimization of DATA DIVISION regardless of optimization levels

INLINE control

COBOL 6.3

GA: 2019

Performance improvements (z15)

New language features: **UTF8**, Dynamic-length elementary items, FUNCTION keyword in REPOSITORY paragraph

64-bit (AMODE 64) Support

COBOL 6.4

GA: 2022

Performance improvements (z16)

New language features: User Defined Functions

New, simplified COBOL/Java interoperability Framework

AMODE 31/AMODE 64 interoperability

Integration with Automatic Binary Optimizer

COBOL 6.5

GA: 2025

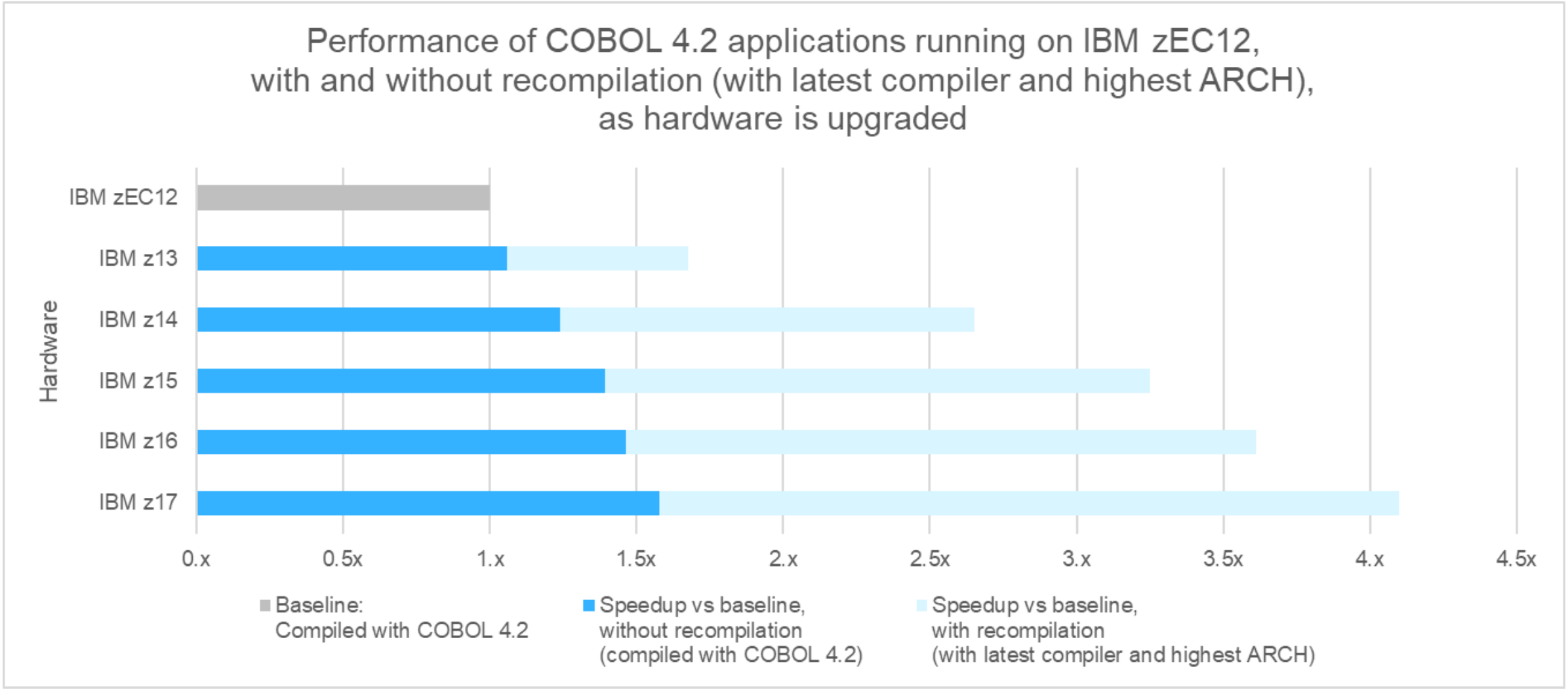
Performance improvements (z16)

New language features: User Defined Types

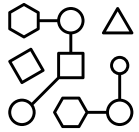
VSAMDB (NoSQL database) support

Integration with Automatic Binary Optimizer

Enterprise COBOL Performance for compute intensive applications

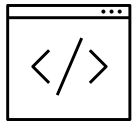


COBOL Upgrade Challenges



Massive amount of COBOL programs without view into inventory

Many customers have thousands of COBOL programs that have accumulated over the past 50+ years, and the size of organizing (taking inventory), prioritizing, and recompiling all of it at once (or even in small pieces) can be massive.



Incorrect COBOL source code

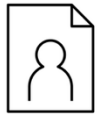
Invalid data: some customers have COBOL programs that contain bad programming practices such as undefined data items (variables).

- Memory layout changes from previous compilers: May alter runtime behavior
- Legacy Reliance on undefined behavior can lead to incorrect results at runtime

Very back-level programs have special challenges

It's very common for customers to be using a mix of versions of COBOL compilers today.

COBOL programs compiled with 70s and 80s compilers may require COBOL runtime updates (to z/OS Language Environment) and source code conversions (to the COBOL 85 language standard)



Skills

Many organizations lack sufficient senior COBOL expertise to drive upgrades. Leveraging junior talent is essential to balance workloads and accelerate modernization.

COBOL Offerings over time

Compiler	Front end	Back End
OS/VS COBOL	74 Std	1st generation
VS COBOL II	85 Std (new)	2nd generation (new)
COBOL/370	85 Std (same)	2nd generation (same)
COBOL for OS/390 V2	85 Std (same)	2nd generation (same)
COBOL for z/OS V3	85 Std (same)	2nd generation (same)
COBOL for z/OS V4	85 Std (same)	2nd generation (same)
COBOL for z/OS V5	**85 Std (same)** **Select 2002 Std**	3rd generation (new)
COBOL for z/OS V6	**85 Std (same)** **Select 2002 Std**	3rd generation (same)

CUAZ
Upcoming Release

CUAZ
Current Release

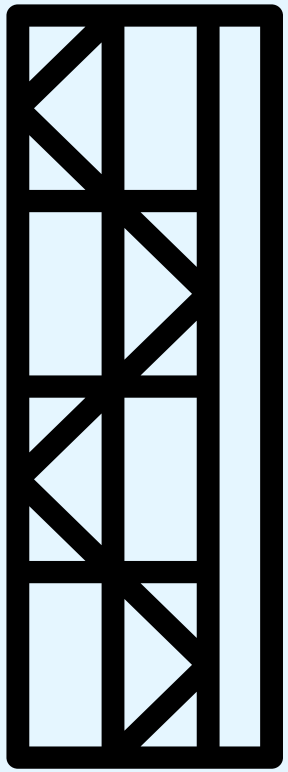
COBOL Upgrade Advisor for z/OS 1.1 (CUAZ)

CUAZ accelerates and simplifies upgrade to Enterprise COBOL 6

Intended for customers upgrading from COBOL 4.2 or earlier.

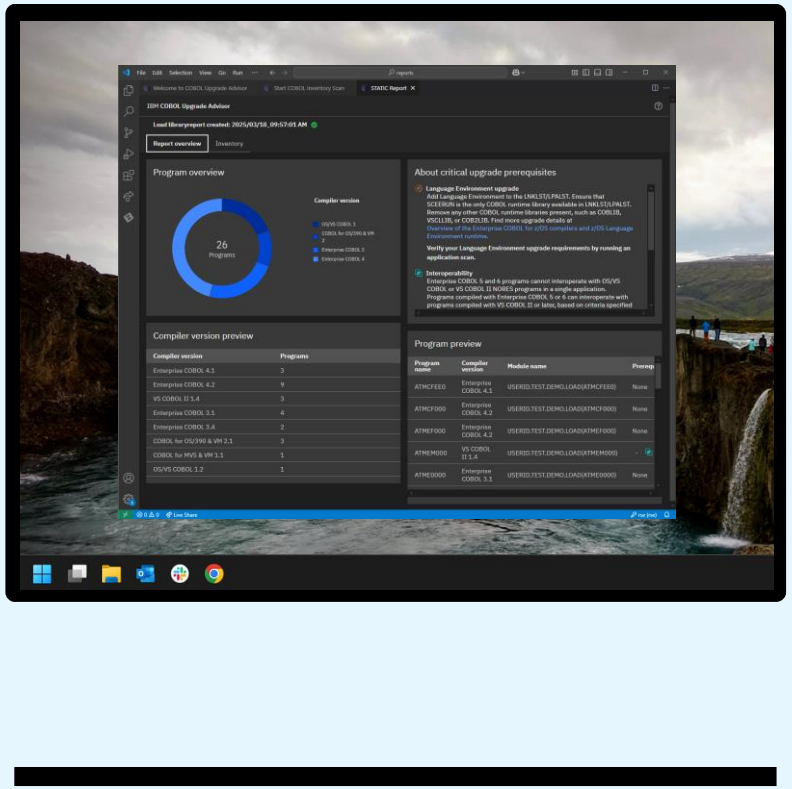
Product has 2 licensed parts:

- Virtual Server
- Authorized User licenses



CUAZ analysis engine running on IBM Z

Zowe connects the mainframe to the workstation



CUAZ output displayed on workstation via VS Code

CUAZ Key Features

Planning insights

- Visual summary of COBOL programs in a VS Code interface
- View compiler version, date of last compilation, compiler options and more
- Judge the size and complexity of your upgrade project
- Export COBOL inventory as CSV

Resolving Prerequisites

List of actionable pre-upgrade tasks per program

- Confirm LE is enabled
- Identify potential interoperability issues

Ensures relevant upgrade tasks are not missed

Recompiling Assistance

- Identifies deprecated options with removal/replacement guidance
- Identifies new compiler options available in COBOL 6, including ones to detect invalid data problems

Run tests and fix invalid data problems

- Identifies invalid data and code problems based on runtime messages, with direct links to the exact source line, making it easy to jump in and fix issues fast

CUAZ Components

Load library scan

Select COBOL load libraries to scan and get a complete view of your COBOL inventory. Use this scan when planning and scoping the COBOL upgrade project across the COBOL inventory, and when you want to upgrade one or more load libraries at once.

Application Scan

This scan shows all the programs involved in the application, including dynamically called COBOL programs that might also need to be updated. Use this scan when you know which specific application you want to upgrade and want to learn how to upgrade the chosen application.

Invalid data detection scan

Identifies invalid data and code problems based on runtime messages, with direct links to the exact source line, making it easy to jump in and fix issues fast

Demo

Why CUAZ?

Benefits

Accelerate COBOL upgrades using junior developers and architects

Improve speed to plan, recompile and test, spend less upgrade project time

Modernized developer workflow using VS Code based extension

Outcomes

Reduction in resources and skills required for COBOL Compiler upgrade

Reduction in cycle times and increase in agility

Standardized compiler upgrade practices across applications and teams

Improved quality of COBOL application by fixing long standing invalid data and program issues

Business values

Optimize application costs to fully utilize the latest IBM Z hardware

COBOL applications built with a supported compiler, reducing security risks and saving MIPS

Grow COBOL expertise while delivering upgrades at a lower operational costs

Use latest WCA4Z features such as Explain, Refactor and Optimize, for better understanding of COBOL application and extending to their capabilities

CIBC's success story about upgrading 40,000 COBOL programs

Business challenge
To maintain the highest service standards for its customers, CIBC had to upgrade business-critical mainframe applications to use the latest IBM Enterprise COBOL for z/OS 6.

Solution
The banking applications are used by both employees at banking centres, and customers. For example, when a banking transaction is performed on a mobile phone, the account balance is automatically updated. An application runs on the mainframe to update the balance in real-time and keeps track of the account balance and fees. A combination of batch applications, online applications (COBOL/CICS), and data management applications (COBOL/IMS, COBOL/Db2) were upgraded to Enterprise COBOL 6.

Transformation
CIBC successfully upgraded 100% or their 40,000 COBOL applications in three years to leverage the latest IBM z15 hardware features and IBM compiler optimization technology available on the bank's IBM z15 mainframe.

Results

- Significant cost reduction in operating costs
- Cleaner code, reducing technical debt
- Avoid the risks of using unsupported compilers, with additional support extension fees
- Skills upgrade for new employees (to learn and understand the functionality of their code base)

"We chose to do this upgrade for currency, stability, and to avoid extended support costs. If you can also get some benefit from reduced CPU consumption, why wouldn't you do that? You're going to eventually need to upgrade anyway. You can't stay in an old compiler forever, so the sooner you move on, the better off you are."
"The upgrade was mostly straightforward."

Murray Schock
Director, Mainframe Shared Services,
CIBC

Link to full client story

IBM COBOL Upgrade Advisor for z/OS: Anticipated roadmap highlights

Today's Capabilities

Automated COBOL Inventory and personalized COBOL insights

Load Library Scan:
Provides a complete view of COBOL inventory

Application Scan:
Application specific scan that provides stats based running that application including CPU, date, time, compiler options etc.

Invalid Data Detection Scan
Automatically identify invalid data and code problems based on runtime messages, with direct links to the exact source line

1H '26

Intended Capability	Outcome
AI powered problem determination and automated code transformation	**Root Cause Analysis for Data Issues** Quickly identify invalid data problems and provide actionable fixes. **Legacy COBOL Syntax Detection** Automatically spot outdated COBOL syntax and provide recommended fixes **Compilation Error Resolution** Parse compilation errors during recompile and provide actionable fixes.

Future

Extended Dynamic Scan - Application architect can detect which applications are running in production, to prioritize them for upgrade

Support online applications – A user can use the application scan to scan CICS, IMS online applications.

CUAZ Agent & MCP Server: A user can automate the steps needed for upgrade including inventory scanning and recompilation.

Continuous mainframe-specific model enhancements / fine tuning plus ongoing improvements to delivered capabilities

Roadmap is subject to change without notice

USE CASES

OBTAIN A COMPLETE VIEW OF THE COBOL MODULES AVAILABLE ON THE SYSTEM

Without CUAZ

Alex

Application manager

"I want to understand our total COBOL inventory."

Resourcing

- Intermediate mainframe developer
- Intermediate COBOL developer or COBOL architect

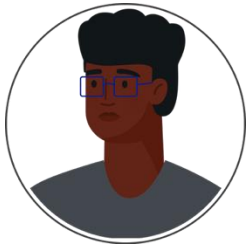
How to

Use a tool such as ABMLIST

1

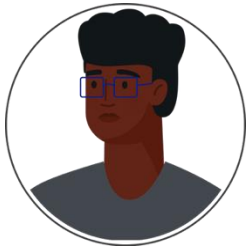
Use AMBLIST to generate a report for all the COBOL load modules and pass it to the COBOL developer.

2



Parse relevant information from the AMBLIST report.

3



Creates a CSV or inventory file and passes it to the manager.

With CUAZ

Alex

Application manager

"I want to understand our total COBOL inventory."

Resourcing

Junior COBOL developer

How to

Use the COBOL Upgrade Advisor(CUAZ) to get a full COBOL inventory.

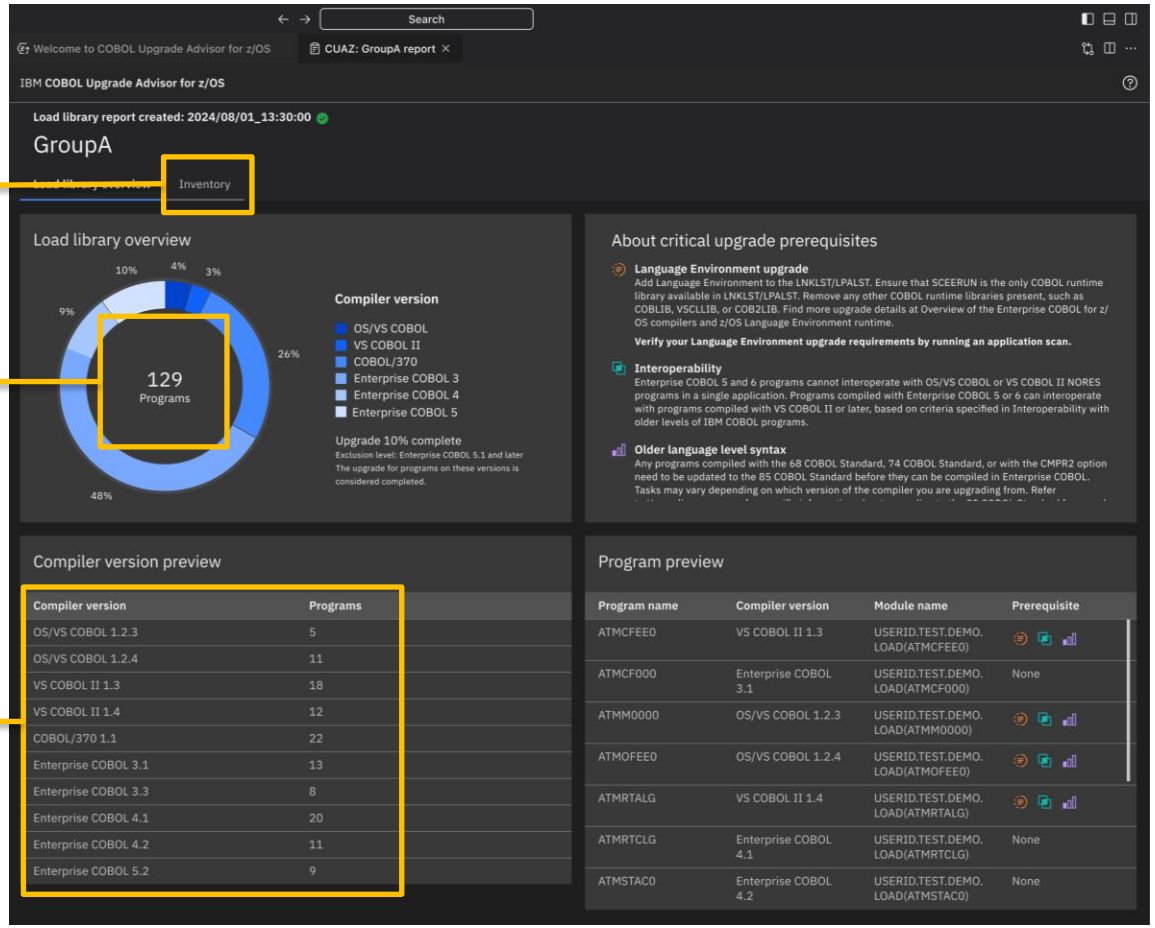
1

Run CUAZ Load library scan

- Full program inventory
- Compiler options
- Compiled date and time

Total COBOL programs

All the compiler versions in use



OBTAIN A COMPLETE VIEW OF THE MAKE-UP OF A COBOL APPLICATION

Without CUAZ

Alex

Application manager

"I want to know which modules this application is calling."

Resourcing

- Intermediate mainframe developer
- Intermediate COBOL developer or COBOL architect

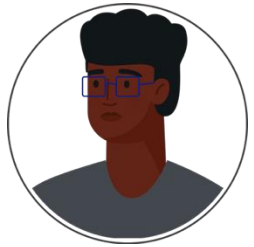
How to

Use a tool such as ADDI to analyze the COBOL source, and AMBLIST or Load Module Analyzer to get the COBOL compiler versions.

1

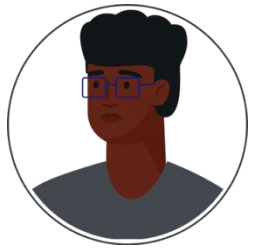
Run a source code analysis tool on the application to get program control flow: Specific knowledge to use a tool such as WCA4Z-Understand

2



Locate the load modules corresponding to the programs that are discovered: basic COBOL developer skills

3



Run a load module analysis tool to get compiler versions

With CUAZ

Alex

Application manager

"I want to know which modules this application is calling."

Resourcing

- Junior COBOL developer

How to

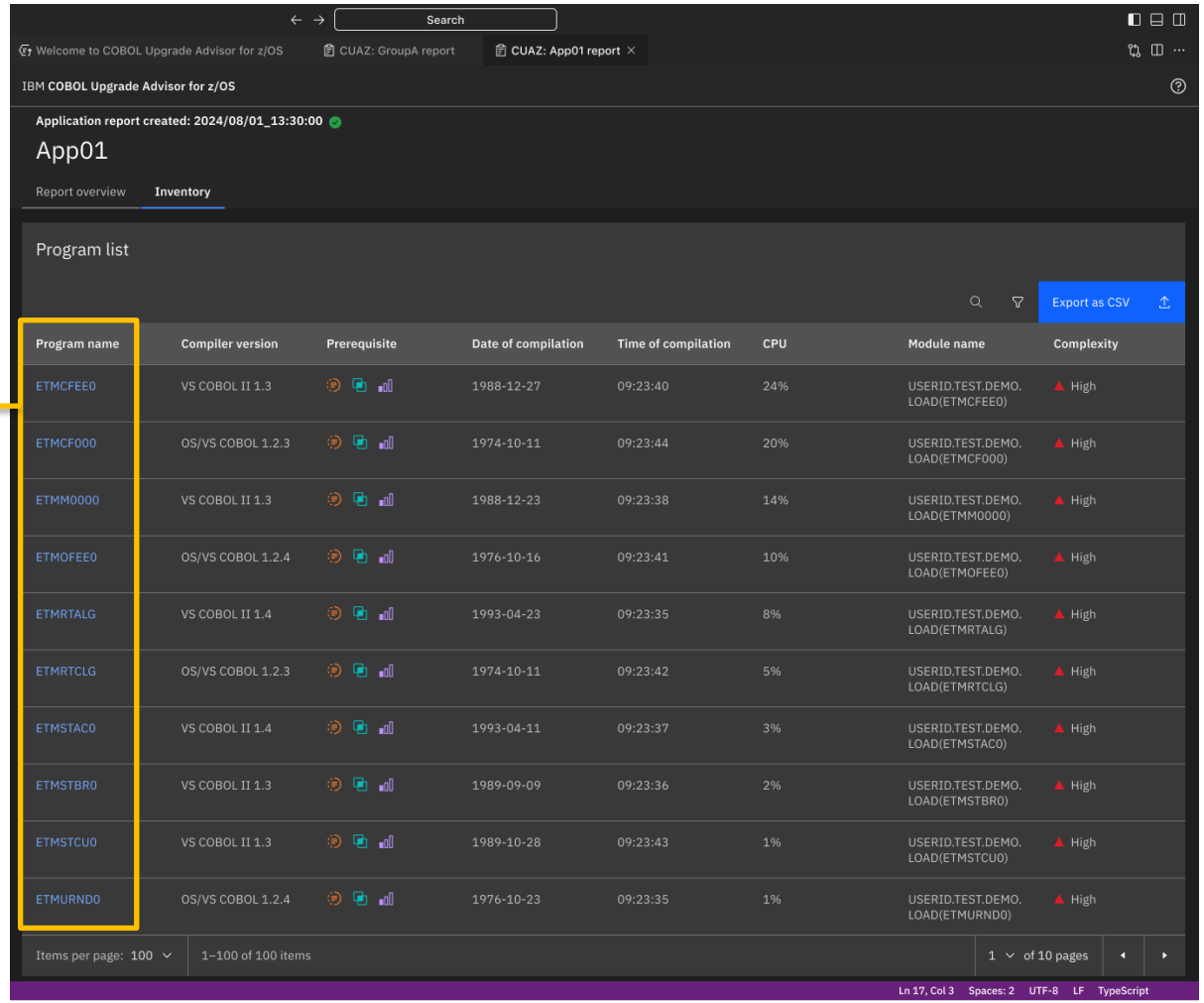
Use CUAZ to:

- Automatically identify all the programs the application is calling.

1

Run CUAZ application scan

List of all the programs that make up the application.



PLAN THE UPGRADE

Without CUAZ

Alex

Application manager

“I want a strategic plan for the COBOL upgrade based on the COBOL inventory.”

Resourcing

- Advanced COBOL developer
- COBOL architect

How to

Consult the COBOL migration guide: A 400-page manual that details tasks required to upgrade to COBOL 6

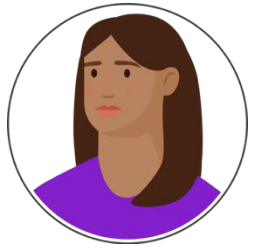
1

Manually determine pre-requisite tasks, such as interoperability or whether the old COBOL language standard is used.

2

Manually create specific upgrade tasks, such as compiler options, to be used for the recompile.

3



Determine the priority upgrade list based on the organization’s needs, using factors such as compiler version and required upgrade effort.

With CUAZ

Alex

Application manager

"I want a strategic plan for the COBOL upgrade based on the COBOL inventory."

Resourcing

- COBOL architect

How to

Use CUAZ to:

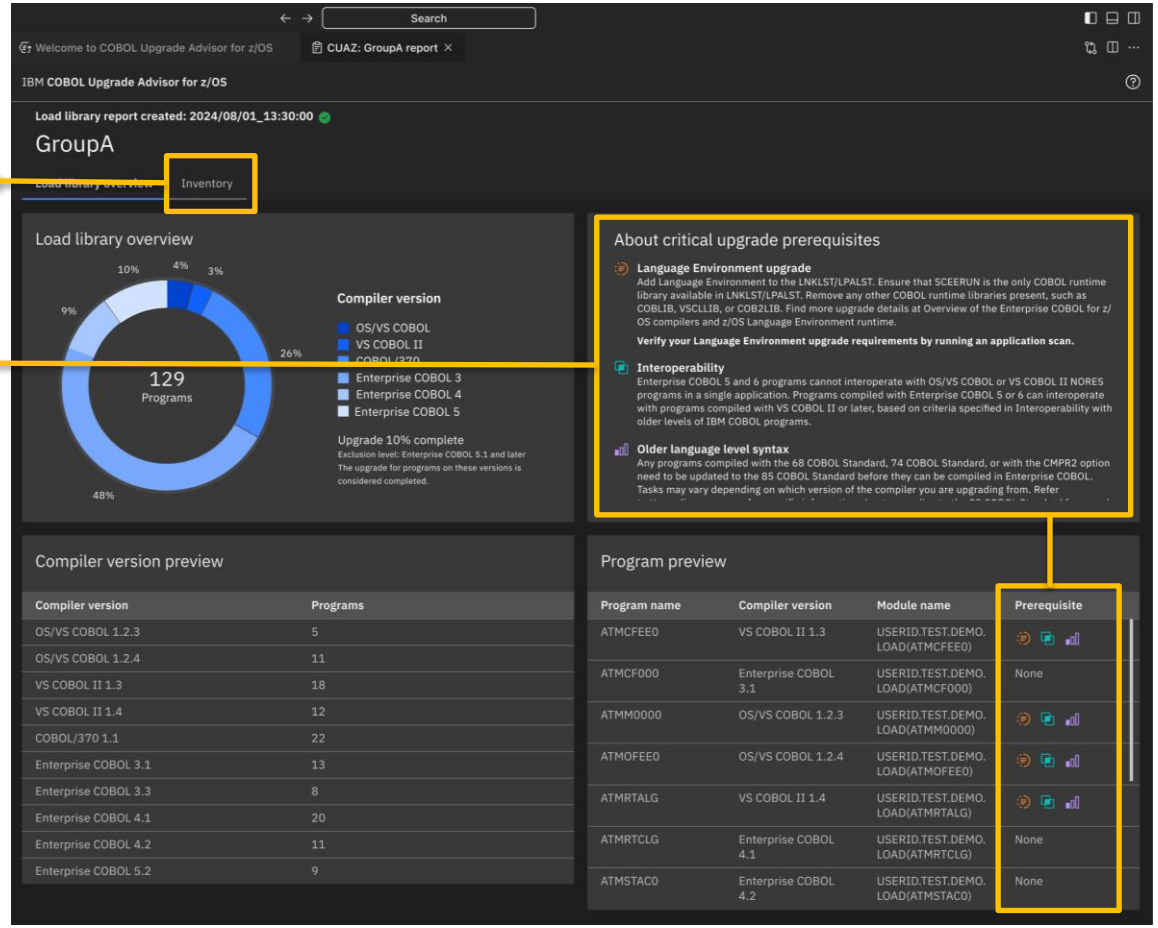
- Automatically identify upgrade prerequisites.
- Get upgrade complexity suggestions per program.

1

Create a priority upgrade list based on the organization's needs by reviewing prerequisite tasks and complexity suggested by CUAZ.

Program complexity level

Upgrade prerequisites



RESOLVE SOURCE-LEVEL ISSUES RESULTING FROM INCORRECT CODING PRACTICES

Without CUAZ

Alex

Application manager

"I want to fix the invalid data problem encountered during the upgrade."

Resourcing

- COBOL tester
- Advanced COBOL developer

How to

Review and address invalid data problems one at a time or create a message report manually.

1

The COBOL tester runs the recompiled application and passes the messages at runtime to the development team

2

Prioritize and address the invalid data issues one at a time by locating the source and documentation on how to fix the issues manually

3

Update the source and hand over to the test team for more testing, i.e., repeat steps 1 and 2 above

With CUAZ

Alex

Application manager

"I want to fix the invalid data problem encountered during the upgrade."

Resourcing

- Junior or intermediate COBOL developer

How to

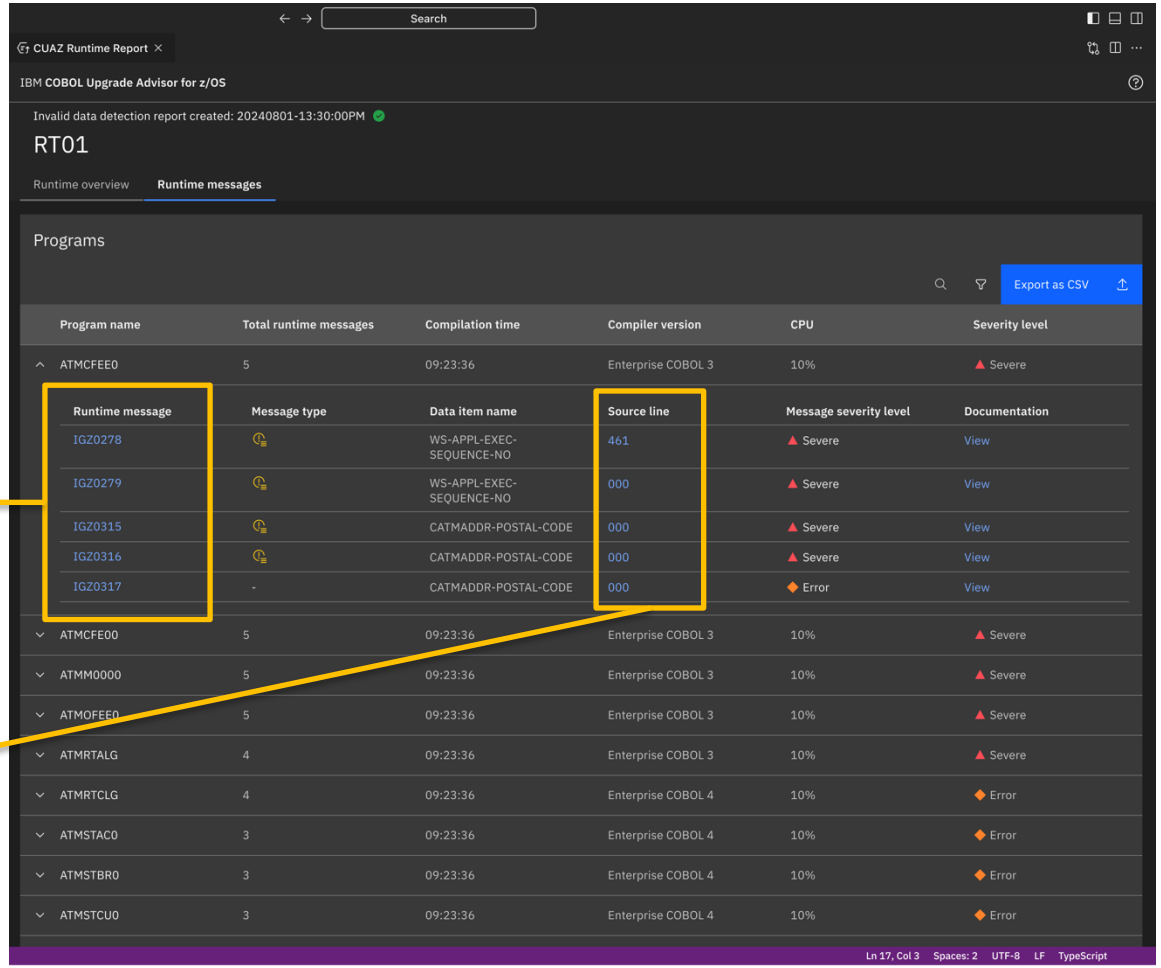
- Automatically parse the Invalid Data Detection report and group the reports either by message type or module
- Automatically open the source file and highlight the code that causes the issue

1

Run an Invalid Data Detection scan, review the runtime message, the report and fix the issues by opening the source file automatically. Repeat this step as needed.

Runtime messages per program

Open a source line



Resources

- IBM COBOL Upgrade Advisor (CUAZ) resources

- Announcement letter: Link
- Product page: Link
- Blogs: Link
- New Feature Ideas: Link
- IBM Documentation – Link

- Demos:
  - Interactive demo – Link

- Demo/POC resources:

- zVA self-serve trial using preconfigured test data. Work with your sales rep.

- 90-day trial. Work with your sales rep.

COBOL upgrade resources

COBOL Upgrade Portal: Link

Includes customer success stories, FAQs, upgrade videos and more

COBOL upgrade services

IBM Expert Labs is ready to help with CUAZ with 2 upcoming offerings to help you with implementation and assistance.

IBM Consulting is ready to help with CUAZ. IBM Consulting has many years of experience with COBOL upgrade services.

IBM Sponsor User program

Shape the future of IBM

What we do

At IBM, we put our users at the **center** of our work to create amazing user experiences.

By **incorporating feedback** from our clients at every step of the design and development process, we build products that solve real user problems.

Together, we'll create **exceptional experiences** for you and the marketplace!

Benefits for You

Connect directly with our Product and Design teams who are eager to hear about your needs and perspectives.

Insider access to early prototypes and confidential information about future product plans.

Influence on our product roadmap and future releases.

How to Join

Let your IBM contact (preferably your design researcher) know you'd like to become a Sponsor User or drop an email to vikash-kumar@ibm.com and they will onboard you.

Or Fill out this form by clicking this link or scanning the below QR code to express your interest:

Your feedback is important!

SHARE
EDUCATE • NETWORK • INFLUENCE

Submit a session evaluation for each session you attend:

www.share.org/evaluation