2026

ISSUE 15

QUARTERLY JOURNAL OF VERYANT AND isCOBOL

veryant NEWS

THIS ISSUE

1. Introducing to isCOBOL 2026R1 - The Future Directions Survey Results 2. PRISM Convention 3. Server Sizing in Production 4. Have You Seen This? 5. Focus on it - G7 - Smart City Taipei 6. AI and COBOL 7. The Versitility of isCOBOL Message Box 8. Last page

Results of our Future Directions Customer Survey

Thank you to everyone who took the oppurtunity to respond to our recent Future Directions Customer survey.

The main reason for the survey was to help Veryant shape our future directions to our customer's needs. The results told us that very few customers use or need the 32-bit versions of our products, and that an up-to-date default WebBrowser control would be welcomed. There was a fairly even mix of Windows and Linux/Unix, as well an even spread of Java versions 1.8, 11, 17 and 21.

Another part of the survey asked where our customers get their communication from Veryant. The release overview for each version was popular, as was the Veryant Newsletter. Lesser used resources included the YouTube videos and the KB articles.

Introduction to isCOBOL 2026 R1

New Features and Enhancements

The isCOBOL 2026 R1 release brings a fresh wave of improvements designed to make your work smoother, faster, and more modern.

Veryant focused this update on helping teams keep their COBOL applications current—whether that means cleaner UIs, stronger integrations, easier debugging, or more flexible deployment options. Here's an overview of what's new in isCOBOL 2026 R1:

NoSQL Bridge: NoSQL Bridge is a stateless REST API layer that exposes isCOBOL supported indexed files over HTTP using JSON. It embeds the isCOBOL runtime to access data files, supports all indexed file formats, and reads EFD dictionaries generated by the compiler (-efd). It ships with a lightweight Jetty based server and can also be deployed as a WAR file (veryant-nsb.war) to external servlet containers such as Tomcat.

Material Design Support: A new MATERIAL-DESIGN property enables Material Design styling for entry fields, bringing a modern look commonly found in web and mobile applications.

Improved Look-and-Feel Integration: The new LAF-COLORS style has been implemented for windows, toolbars, and ribbon controls, simplifying color management for applications that follow the default Look and Feel color scheme.

We're Heading to the PRISM Conference in San Diego!

We're excited to share that Veryant will have a booth at the 2026 PRISM Conference in San Diego, taking place May 17–20, 2026 at the Sheraton Hotel and Marina.

What is PRISM?

PRISM (Public Retirement Information Systems Management) is a professional organization that brings together members who work with public retirement systems to share insights, best practices, and technology innovations. Its annual conference features keynote presentations, breakout sessions, and collaborative discussions that help attendees exchange lessons learned and explore new approaches for their organizations. [prism-assoc.org]

PRISM
PUBLIC RETIREMENT INFORMATION SYSTEMS MANAGEMENT

If you'll be attending, we'd love for you to stop by our booth. Come say hello, learn what's new with isCOBOL 2026 R1, and connect with our team and others in the community.

isCOBOL Runtime Enhancements The runtime now includes updated versions of several third party libraries to improve security and modernize deployed applications. A new function, C$PROGINSTACK, has been introduced to check whether a program is currently in the call stack.

Greater COBOL Dialect Compatibility: Compatibility with other COBOL dialects—such as Micro Focus COBOL and IBM COBOL—has been enhanced. The compiler now supports additional functions and new ESQL statements.

Debugger Improvements: The isCOBOL 2026 R1 debugger now allows setting breakpoints while the program is running.

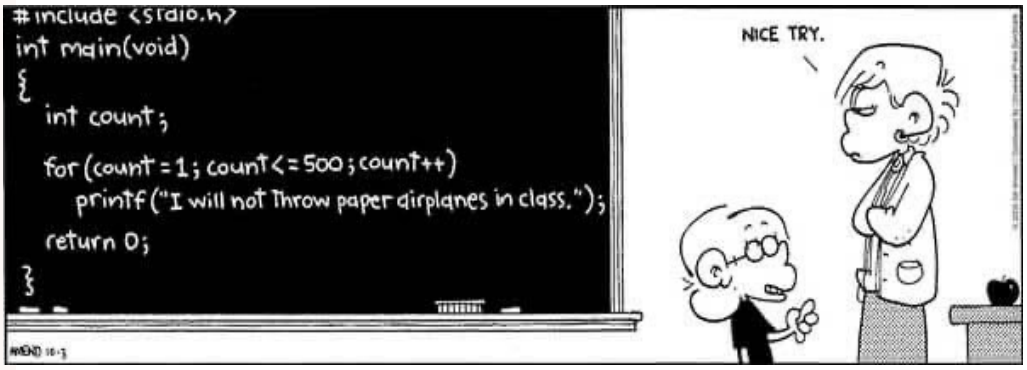
isCOBOL Server Enhancements: The isCOBOL Server Panel can now generate thread dumps and heap dumps directly, improving diagnostic capabilities.

Database Bridge: Two new properties allow automatic creation of an additional column on Oracle databases to store a file's parent directories.

Maven Integration: The official isCOBOL Maven plugin now enables compiling COBOL sources directly as part of Maven build lifecycles.

isCOBOL WebClient Updates: A new window style, IWC-DOCKABLE, allows windows to be marked as dockable in web environments.

c-treeRTG v5 Enhancements: isCOBOL users can now increase the record length of an indexed file and add new keys without needing to migrate existing records.



Determining Server Sizing isCOBOL Applications in Production

Getting your server sizing right before going live can mean the difference between a smooth deployment and costly performance problems down the road. The good news is that with the right approach, it's a straightforward process.

When going live, having the right hardware is critical to ensuring a smooth experience for your users. Hardware sizing isn't always straightforward — it requires balancing cost against performance. While "bigger is better" generally holds true, overpowered hardware is an unnecessary expense if your application and its workload are not resource-intensive. Conversely, underpowered processors, insufficient RAM, or small disks can cause slowdowns or hangs when the application uses a large number of resources or when a high number of users are connected and working simultaneously. The guidelines below will help you choose the right server hardware for your isCOBOL application

Our scenario: A thin-client application using c-tree as the database, deployed on Windows with both the isCOBOL Server and c-tree Server on the same machine. How do you size this server?

The two most critical resources to plan for are **disk space** and **RAM** — the two that, when exhausted, can halt your application or operating system entirely.

Sizing Disk Space

To determine the required disk space, account for the size of your software, your initial data, and projected data growth over time. You can measure current software and data sizes on your development machine by checking the installation directories for Java, isCOBOL, and c-tree. For example:

- Java JDK: 290 MB
- isCOBOL SDK: 170 MB
- c-tree software: 1 GB
- c-tree data files (initial state): 50 MB
- **Total: ~1.5 GB**

From there, factor in data growth. If you expect data to grow by 500 MB per year and the application will be in use for at least five years, add 2.5 GB — bringing your total to roughly **4 GB** as a practical starting point.

Sizing RAM

Windows Task Manager is your best tool here. Start by recording the RAM usage of both the isCOBOL Server and c-tree Server processes while no clients are connected — this is your baseline.

Next, launch a thin client, run the application, and perform typical operations while monitoring RAM usage for both server processes. The difference between peak usage and your baseline represents the RAM footprint of a single user.

"The two most critical resources to plan for are disk space and RAM"

Because all concurrent users could be performing resource-intensive operations simultaneously in a live environment, multiply that per-user figure by your expected maximum number of concurrent users. For example, if one user consumes 200 MB and you expect up to 50 concurrent users, plan for 10 GB of RAM — plus your baseline overhead for the isCOBOL Server and c-tree Server processes.

By correctly sizing your disk space and RAM before deployment, you can ensure a smooth rollout, keep performance issues at bay, and avoid costly hardware problems down the road.

Have You Seen This?

Newest Video:

Setting User Info in AS Admin Panel
WebClient Basics
What's new in 2025 Release 2
Thin Client Login Setup

New KB Articles:

How to use the Application Server's login screen for your application
Creating a custom login screen for thin client users
Using DatabaseBridge with Multi-Company / Multi-Year ISAM File Structures
Managing High-DPI Behavior in isCOBOL UIs (Windows and Multi-Platform Options)

FOCUS ON IT

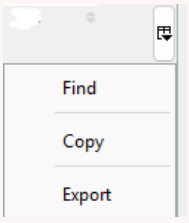
Easiest way to export Grid control contents to Excel

The grid control can export its content to an Excel file without requiring any additional coding.

To enable this feature, simply add the Heading-Menu-Popup property to the screen section

```
screen section.
...
  03 screen-1-gr-1 Grid
     line 15.5
     column 2.6
     size 28.4 cells
     lines 18.6 cells
     id 3
     no-box
     heading-menu-popup 9
     column-headings
     row-dividers 1
     heading-color 133
     heading-font Default-Font
     cursor-frame-width 3
     num-rows 5
     .
```

This property adds a small menu icon in the upper right corner of the grid. When you click the icon at runtime, you'll be prompted to choose where to save the exported Excel file.



Read the details in isCOBOL's UI Documentation here.

G7

The G7 event in Rome has come to a close, having been held from October 15–17, 2025. We extend our sincere thanks to all participating companies from Brazil, the Philippines, Japan, Thailand, Taiwan, and the United States for their valued contributions.

Our discussions centered on Artificial Intelligence in programming languages and Digital Transformation.

We look forward to our next gathering in September 2026 in São Paulo, Brazil.

.

SmartCity event in Taipei

Veryant was happy to be part of the Smart City Summit & Expo (SCSE) 2026, held March 17–20.

As Asia's largest annual smart city tradeshow, SCSE—established in 2014—serves as a leading platform for the Information and Communication Technology (ICT) to showcase cutting edge Internet of Things (IoT) and Artificial Intelligence (AI) solutions that advance urban development.

We were proud to exhibit at Booth P214a alongside our partner, SYSCOM.

AI and COBOL: More Relevant Than You Think

The current wave of AI excitement can feel distant if you're an open-systems COBOL developer. Most of the headlines focus on mainframe COBOL— leaving off-mainframe COBOL shops wondering whether any of this applies to them. It does. AI offers meaningful benefits to COBOL developers regardless of platform, and those benefits are available right now.

How AI Supports Developers

At its most practical, AI helps developers work faster and more efficiently while improving the long-term health, maintainability, and scalability of existing codebases. It can handle repetitive tasks, find deep-buried bugs, and serve as an on-demand co-developer, mentor, or subject-matter expert.

Choosing the Right Tool

General-purpose AI tools — ChatGPT, Google Gemini, and Microsoft Copilot Chat for instance — are a reasonable starting point, and our research suggests some outperform others for COBOL-specific work. Claude, for instance, performed well on several practical tasks: converting nested IF statements to cleaner EVALUATE logic, handling complex date and mathematical calculations, and generating ESQL.

That said, the more powerful option for most developers will be an **AI coding assistant**: a plugin (or standalone IDE) that operates directly within your development environment, with full visibility into your codebase as you work. Claude Code and GitHub Copilot are strong examples of this category, and isCOBOL's VS Code extension integrates well with both. SourceForge has a page listing COBOL AI Coding Assistants that is constantly being updated.

AI coding assistants can:

- Generate, refactor, and document code
- Search and analyze code for bugs, dead code
- Identify and fix errors in real time
- Automate test generation
- Replace hard-coded values with configuration settings
- Break up large, complex code blocks
- Assist project planning and task breakdown
- Help developers from other language backgrounds learn COBOL
- Write Java code callable from isCOBOL

> ***"For COBOL shops willing to invest the time, AI represents a meaningful shift in how development work gets done."***

For COBOL shops willing to invest the time, AI represents a meaningful shift in how development work gets done — provided it's approached with the right expectations.

Using AI Wisely

AI is a powerful tool, but only when used correctly. When developers know how to work with it effectively, AI can accelerate development, improve code quality, and take repetitive tasks off their plate. Used poorly, however, it can introduce bugs, generate incomplete or outdated code, and — if its limitations aren't understood — become a time sink rather than a productivity boost.

The difference often comes down to training and technique. Developers who know how to ask the right questions get useful, actionable output. Those who don't may find themselves reviewing and discarding more code than they write.

Security is another consideration that's easy to overlook. Most general-purpose AI tools are cloud-based, which means code pasted into them may be transmitted to and stored on external servers. Before using any AI tool with proprietary or sensitive code, developers should understand their organization's data privacy policies and review the tool's terms of service. AI coding assistants that run locally or within a controlled environment are worth prioritizing for shops where code confidentiality is critical.

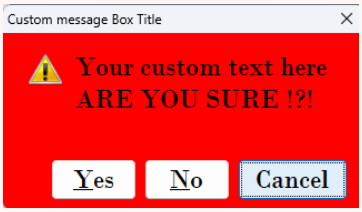
The Versitility of isCOBOL's Message Box

The DISPLAY MESSAGE BOX statement provides several optional clauses for customizing your message box layout -- from icons to fonts and beyond.

If the built-in clauses don't meet your needs, you can implement your own message box as a GUI program with a floating window, replacing the runtime default entirely. In short, you have full control over how message boxes look and behave.

For full details, refer to the documentation here.

See a sample program in $ISCOBOL/sample/issample/s-gui/MESSAGE-BOX-CUSTOM.cbl



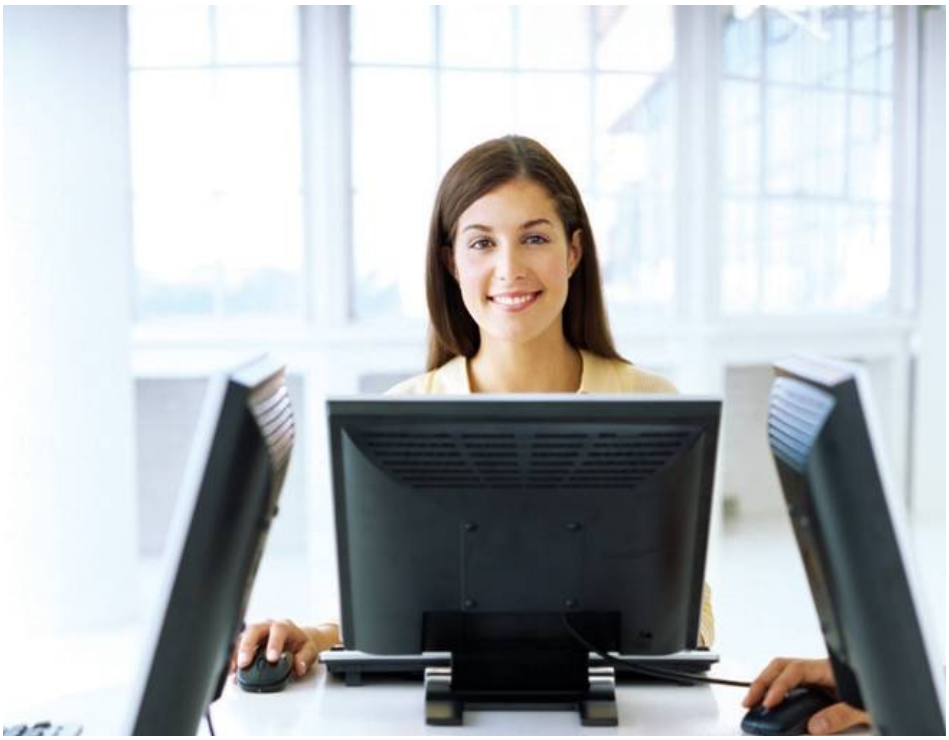
AI is still a maturing technology, and getting real value from it requires experimentation. The more it's used, the better it becomes — but in COBOL development especially, where codebases often span decades and multiple developer generations, human oversight remains essential. Complex thinking, architectural judgment, and deep institutional knowledge can't be delegated to an AI tool, no matter how capable it becomes.

The goal is to use AI to its fullest while keeping those irreplaceable human skills at the center of the work.

Invest in learning how to use it well, stay aware of its limitations, and always review what it produces. The payoff is real — but so is the cost of getting it wrong.

AI won't transform your COBOL development overnight, but the developers who start experimenting now will be the ones best positioned to benefit as the technology matures. The first step is simply getting started.

> ***"In the creative disciplines, L.L.M.s take away the most soulful human parts of the work and leave the drudgery to you" but "in coding, L.L.M.s take away the drudgery and leave the human, soulful parts to you"***
>
> Coding After Coders - The New York Times..



Evolution, without revolution

Contact Us

For supported customer email us at **support@veryant.com**

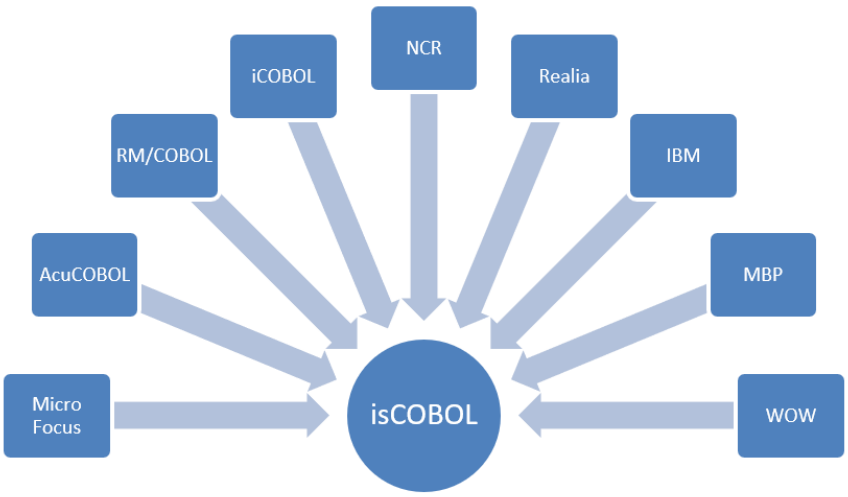
If you would like Veryant to contact you to schedule a technical product briefing, email us at **info@veryant.com**

If you would like Veryant to contact you for special quote or sales assistance email us at **sales@veryant.com**

Corporate Headquarters
6390 Greenwich Dr., Suite 225
San Diego, CA 92122 - USA
Tel (English): +1 619 797 1323
Tel (Español): +1 619 453 0914

European Headquarters
Via Pirandello, 29
29121 - Piacenza - Italy
Tel: +39 0523 490770
Fax: +39 0523 480784
emea@veryant.com

As always, the newest isCOBOL Evolve release contains multiple compatibility additions – as we continue to make your conversion process as smooth, quick, and pain-free as possible.



veryant.com

Follow **Veryant** on

veryant.com