



QUARTERLY JOURNAL OF VERYANT AND isCOBOL



Calling all Beta testers! The next release of 2024R2 is scheduled for the end of September, but you can get a sneak peek now by registering to be a beta tester.

HOW DO YOU PARTICIPATE ?

Send an email to beta@veryant.com to request license keys for the products you plan to test, as well as access to the Beta download site. .

Try it now. It's included in isCOBOL Evolve, both the SDK and the IDE.

Share your feedback with Veryant at beta@veryant.com



veryant

NEWS

THIS ISSUE

1. 2024 R2 Beta **2.** Debugging Part 1 - 32-BIT or 64-BIT: When bitness matters? **3.** Have you seen this? **4.** 2024 R2 BETA has Something New For Everyone **6.** Last page

2024 R2 Beta is open! Extend your Reach with 2024R2 isCOBOL

The upcoming isCOBOL 2024R2 release refines the previous version, introduces a new incremental compiler with support for boolean values, has an updated debugger and a new CHIPS-BOX control. You can get a head start by downloading the beta and testing your application with it.

isCOBOL Evolve 2024R2 introduces an incremental compiler to speed up compilation time for large projects, adds support for boolean data items, and improves the OOP syntax by allowing the VALUE clause on object reference data.

isCOBOL Debugger 2024R2 supports full access to data items of programs and classes in the stack and adds enhancements to Perform stack view and Breakpoints.

isCOBOL 2024 R2 introduces a new control named CHIPS-BOX, a container that shows a list of chosen options in a box, improves the W\$BITMAP library routine and lets developers customize and centralize window creation and control events.

The updated Database Bridge tool has been enhanced to optimize interactions with Microsoft SQL Server.

Are you running an older version of isCOBOL? We're here to help you upgrade to stay with a supported versions and take advantage of new technology. Check out our [end-of-life policy here](#).

Debugging for All Environments

Debugging Cobol can be an... experience. Fear not, intrepid programmers! With a little isCOBOL Debugger know-how, you can transform your code from cryptic culprit to computational champion.

In graphical environments you can simply add a `-d` to your compile and runtime lines using `isccrun` or `isclient`. Let's look at how to debug a program that is running on an operating system without any user interface available. In this environment you'll need to use isCOBOL's Debugger in remote debugging mode.

Here are the basic steps to activate the remote debugger:

- 1) Compile the program(s) to be debugged with either `-d` option or `-dx` option
- 2) Set the `iscobol.rundebug` * property in your run environment to either value 1 or 2:

- `iscobol.rundebug=1`, the first program will start and run immediately. The remote debug session waits and listens for a program that is compiled for debug to launch, connecting to the program when it detects it. This is useful when you only want to debug some of your application's programs.
- `iscobol.rundebug=2`, the runtime framework will start in debug mode, pausing to connect to a remote debugger before running the program. This is useful when all your programs are compiled for debug and you want to start your remote debugger at the very first line of the application.

```

valerio@ubuntu: ~/veryant
valerio@ubuntu:~/veryant$ iscc -d IO-INDEXED.cbl
valerio@ubuntu:~/veryant$ isccrun -J-Discobol.rundebug=2 IO_INDEXED
Debugger listening on port 9999 ...
  
```

- 3) Start the remote debugger session on your desktop.

To start a remote debugging session, the isCOBOL Debugger needs to know the host name of the machine running the program to be debugged and the port number dedicated to the debugger connection. The command is `isccrun -d -r <HostName> <Port>`

```

Command Prompt - isccrun -d X + v
C:\Veryant\isCOBOL_Evolve2024R1\sdk>isccrun -d -r 192.168.26.128
  
```

32-BIT OR 64-BIT: WHEN DOES BITNESS MATTER?

Java classes are platform and bitness independent. For instance, you can compile on Windows with JDK 32bit and run on Linux with JRE 64bit.

Bitness matters when accessing Native Libraries

But there are cases where bitness matters. If your COBOL application calls a `.dll` written in another language, use the bitness of that `.dll`

If your COBOL application calls a C library, the isCOBOL and Java need to match the C library's bitness. If your app also passes pointers as parameters between the COBOL and C, you should also use the `-cp` compiler switch. If the C libraries are 64-bit, you should also use the `-d64` compiler switch.

Where isCOBOL Stands

isCOBOL Application Server and isCOBOL Thin Client don't need to match bitness. You can run a 64-bit isCOBOL server process and access with 32-bit and/or 64-bit clients.

When you use WebClient, you need to use Java 64-bit, WebClient is the only Veryant deployment product that doesn't support 32-bit.

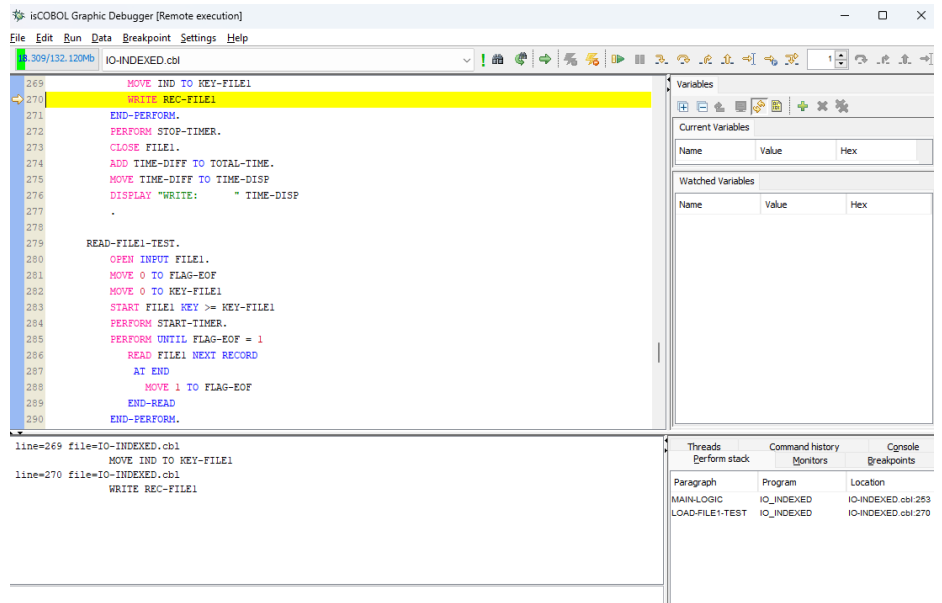
isCOBOL's IDE requires 64-bit Java to run, but can compile your code with Java 32-bit using the integrated Remote Server added in 2021R1.

PLEASE JOIN US ON

Twitter, LinkedIn, or Facebook to up-to-date with Veryant's news



Watch our demonstration videos and Subscribe to our YouTube Channel



In the next newsletter we'll see details of some amazing functionalities available with the isCOBOL Graphical Debugger and in which other environments it can be used. Stay tuned!

Have You Seen This?

We've been busy updated our KnowledgeBase articles. Here's some you might like to revisit:

[What can I use to perform an email validation during entry-field accept?](#)

[Does isCOBOL provide Graph modules to show statistics?](#)

[Does isCOBOL support Crystal Reports or other Report Writers?](#)

[How do I retrieve the version of the Java Compiler and the isCOBOL Compiler that produced a given class file?](#)

[What does the error "Software Incompatibility" mean?](#)

[How do I set up isCOBOL to access Vision files?](#)

[We plan to replace our CICS screens with Java dialogs. How will the COBOL structures be passed between the Java dialogs and the other COBOL programs?](#)

[How can I read an Excel spreadsheet from COBOL?](#)

2024 R2 Beta have something new for everyone

The next release 2024R2 is scheduled for the end of September and the Beta period has started.

Here's a summary of the changes and features in the new version.

Compiler improvements

- incremental compilation under new option `-incr`. Incremental compilation means optimized and faster large compilation tasks with logic to compile only the programs that have been modified since the last compile
- store custom info in the compiled class under new option `-brand`

Compatibility improvements

- more DB2 compatibility in Embedded SQL code
- more IBM and MF COBOL compatibility with support for Boolean data item declaring PIC 1 and USAGE BIT. Functions INTEGER-OF-BOOLEAN and BOOLEAN-OF-INTEGER to manage Boolean data items
- more RM compatibility supporting P\$SETBOXSHADE printing routine

Debugger improvements

- full access to data items of programs/class in the Stack. Support for new option `-c` in all Debugger commands that refer to data items to set the program name or classname:>methodname. All dialogs for data-items need a new field to easily set the new option
- entry-point and section names in the "Perform stack" view and Infostack command
- enhanced breakpoints on program, method and paragraph level stored in the `.isd` file to keep them also after recompilation with line changes.

GUI features

- new control: CHIPS-BOX. It's a container where chips can be added and removed. Events CMD-CLICKED and MSG-CLOSE when the user clicks on the chip text and X icon
- new hook feature on windows to inquire/modify attributes before or after the creation under new configuration `iscobol.gui.window.hook`
- new property WINDOW-STATE to inquire if a window is minimized or maximized

2024 R2 BETA has Something New For Everyone

- more info returned during MSG-MOUSE-ENTER event on complex controls like grid or tree-view to better identify where the mouse is
- create bitmaps from text with the new op-code WBITMAP-TEXT-BOX in W\$BITMAP library routine
- new WOW routines to optimize AS communication in ThinClient. WOWSTARTBUFFERING and WOWSTOPTBUFFERING to start and stop the bufferization of WOW calls and update the video

DataBase Bridge features

- support for light cursors in EDBI routines for Microsoft SQL Server. Runtime configuration iscobol.easydb.sqlserver_row_limit to specify the cursor size.

Profiler utility improvements

- new configurations iscobol.profiler.enable=false to start the program with profiler disabled and iscobol.profiler.elapsed_time=n to set a timeout in seconds on profiler flush

To see more details, [read the isCOBOL 2024R2 release overview here.](#)





Evolution, without revolution



Contact Us

For supported customer email us at support@veryant.com

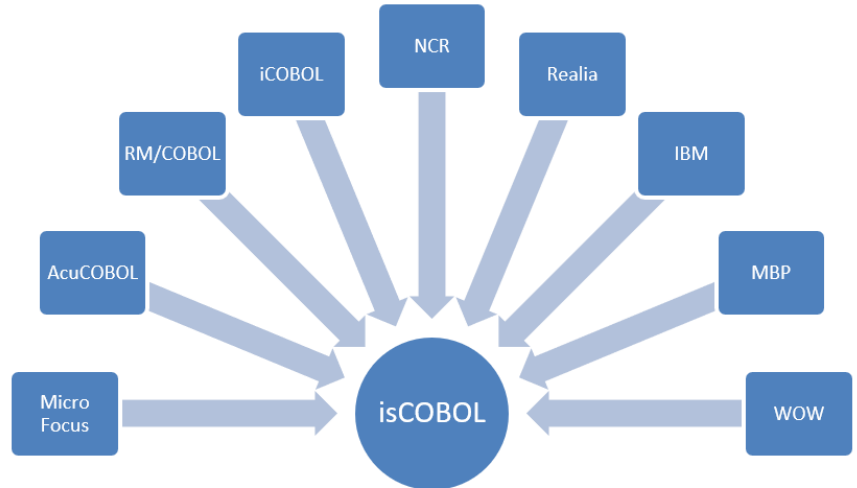
If you would like Veryant to contact you to schedule a technical product briefing, email us at info@veryant.com

If you would like Veryant to contact you for special quote or sales assistants 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

©2024 Veryant - All Rights Reserved