2020/04

ISSUE 02

# veryant NEWS

## Happy 60th Birthday to COBOL. You never looked better!

## WE'RE SO EXCITED ABOUT THE NEW ENHANCEMENTS IN 2020R1.


You can

- insert new GUI controls into your program like SCROLL-PANE, and TREE-VIEW TABLE-VIEW style
- have finer control on the user interface and the management of mouse events using NOTIFY-MOUSE style supported on all controls
- show page thumbnails on print preview, allowing the user to quickly jump to a specific page in a multi-page report

Maurizio Girometti, our CEO, says Veryant was started with a mission to "revitalize COBOL and transform it into a cutting-edge technology language" and with the enhancements in 2019R2 and 2020R1 we're continuing with that mission.

## THIS ISSUE



WHAT'S NEW

# 2020R1 is here

**Veryant is pleased to announce the latest release of isCOBOL™ Evolve, isCOBOL Evolve 2020 R1. isCOBOL Evolve provides a complete environment for the development, deployment, maintenance, and modernization of COBOL applications**

Starting with the 2020R1 Release, isCOBOL is introducing features with Enterprise users in mind: isCOBOL Code Coverage and isCOBOL Unit Test. These features have been available in other languages, such as Java, and now they're available to isCOBOL developers as well. They will help you produce better quality tests for COBOL applications.

These new features are also integrated into the IDE, as well as customized code folding. We've added a few new GUI controls, a scroll-pane and table-view style to our tree-view, that will help you keep your applications looking fresh and up-to-date.

Other improvements include finer mouse control handling, print preview thumbnails, and of course the constant ongoing improvements to performance and compatibility.

This edition of the newsletter will include information about 2019 R2 as well. Normally major new features are introduced in the R1 versions at the beginning of the year, but there are some changes in 2019 R2 that are pretty exciting, like LM-ZOOM to resize the whole screen with one variable, and code-insertion to apply a control style using the configuration file instead of changing your code.

# COBOL TODAY

**Maurizio Girometti, CEO of Veryant, talks about his experiences with COBOL and how it fits into today's world**

*Experience the power of COBOL in the modern world!*

**I started** writing COBOL applications in 1982 on a Texas TI-990 minicomputer. It was a great experience after practicing with Assembler, Fortran and Basic. I found COBOL easy, stable and of course absolutely suitable for business applications.

**With the advent** of new languages, COBOL started having a bad reputation in the IT community. In the beginning there was the 4GL languages illusion: in theory any programmer was able to write complex applications in minutes. *Not really true*.

**Then** other languages promised to deliver cutting edge technology and actually they did. This definitely contributed to increase the bad reputation of COBOL since most of the IT world thought that at this point COBOL was just relegated to old mainframe applications.

*So, is it true that COBOL is old and does not support the new technologies or is this just an "urban myth"?*

**In 2005** we decided to start up Veryant with the purpose to revitalize COBOL and transform it into a cutting-edge technology language, since there is still plenty of software in the world written in COBOL. As a result, isCOBOL® was born and today it is utilized worldwide by thousands of developers.

**So**, assuming that COBOL is an old language is an "urban myth", since isCOBOL provides all the functionalities provided by modern languages, including a graphical user interface, web services, remote thin client, web interface, OOP, a powerful IDE based on Eclipse and much more.

*"In 2005 we decided to start up Veryant with the purpose to revitalize COBOL and transform it into a cutting-edge technology language."*

**And, all this** is portable from mainframes to mobile devices without changes, since at the end of the compilation process what you get is a Java object. This means that if necessary, you can interact with any other Java object, easily taking advantage of the power of the Java world.

**If you want to try** our product, please call our offices and we will be glad to provide more technical and sales information, and deliver an evaluation copy of our products. Our technical staff will assist you during the entire evaluation period.

# Code Coverage and Unit Testing

The new Code Coverage feature will tell you how much of your code is actually getting executed during a run. This is helpful when you're setting up a test environment and want to be sure you're testing as much of the code as possible.

**We've made it easy** to implement (add the -coverage to your run command line) and understand the results with a "Coverage global report" using graphs to show the coverage at the program and paragraph levels, as well as give you access to your code, color-coded to show line-by-line what was executed and what was not.

**We know** not everyone has been able to create an automated test environment, and we've added something to help you with that too. The Unit Test feature lets developers create automated test suites, designed to test that sections of code execute as intended. The goal of unit testing is to isolate sections of a program and ensure that they are working correctly.

**To set up** this testing, you would add an ASSERT statement to the parts of your code you want to test, similar to this:

```
assert string1 = "my string"
otherwise "Test string
manipulation: Error"
```

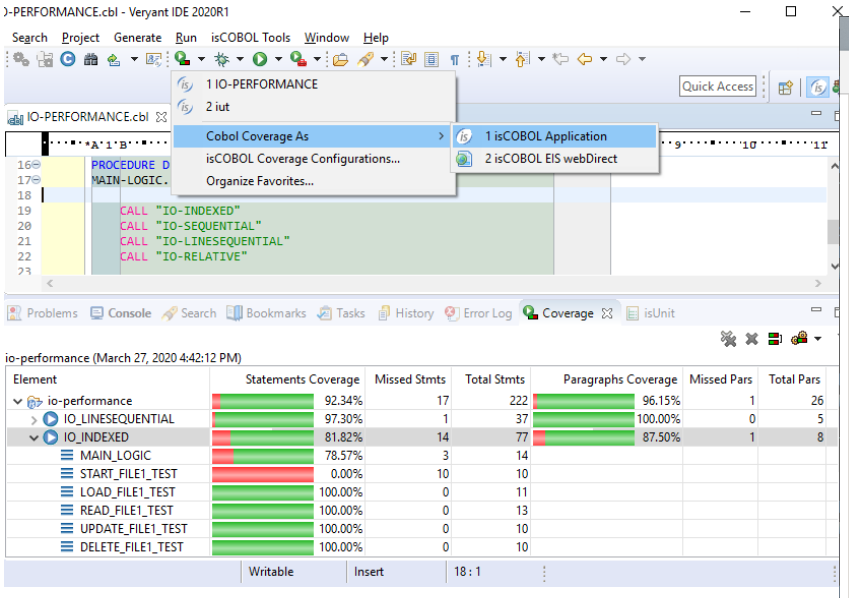
When you compile and run your program with the right switches, the Unit Testing is activated, without these switches the statements are ignored. The Unit Testing feature generates reports similar to the Code Coverage, and when run together the reports are combined.

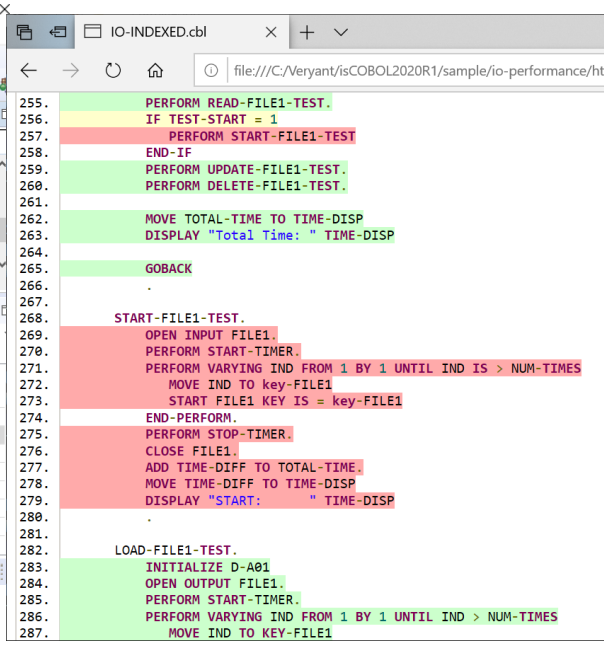
**And the best part?** All of this is integrated in the IDE as well. Contact support or your sales representative for more information or a demo.

## MULTI THREADING SAMPLES

Have you looked at our Multi-threading samples lately? It's an interesting and easy way to add value to your application. The samples show you how to:

- run a clock in the background
- prevent closing one window before another
- make sure the user doesn't run a program more than once
- count the programs running
- run multiple programs in the same or separate environments
- let the user interrupt a running process
- run a progress bar



## DOCUMENTATION HIGHLIGHT

### New Documentation

The isCOBOL Documentation has been redesigned with a new modern look and feel and is now accessible to developers from mobile devices.

This also resolves the problem you may have noticed in the 2019 documentation – Some browsers (like Chrome and Firefox) would not display your installed documentation because of a new security feature they introduced.

Try it out and let our documentation department know what you think. Just click on the email icon in the bottom right of each page.

You can access it **here** to see.

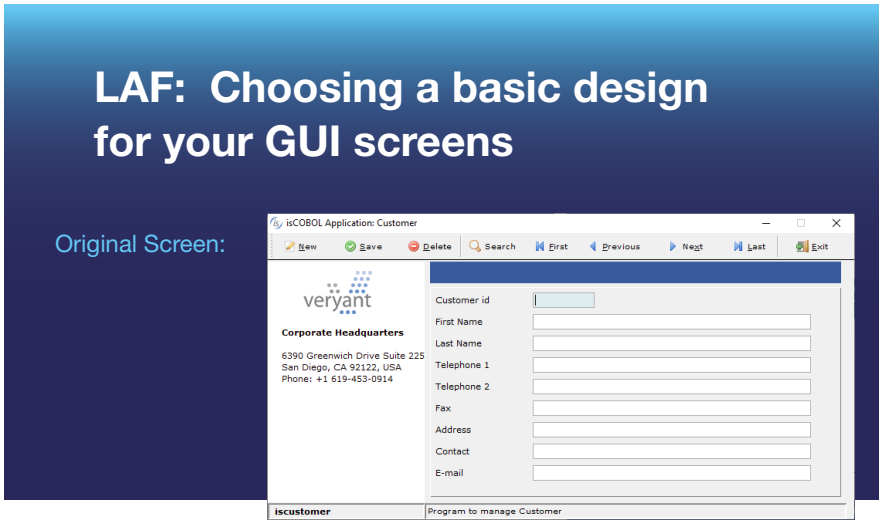
## Have You Seen This?

### New YouTube videos:

- REST/SOAP Webservices
- Installing isCOBOL SDL for beginners
- isCOBOL 2019R1 new features

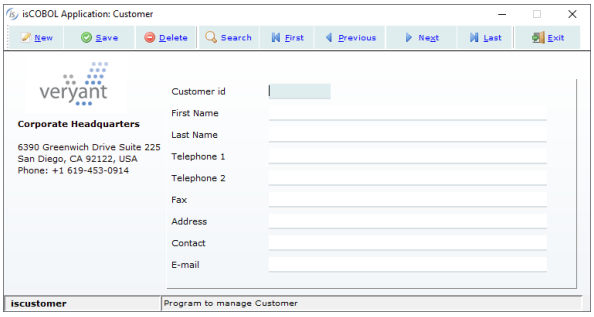
### New Knowledgebase articles:

How to use exisiting FD/SL copybooks in the Screen Programs without generating FD/SL from IDE Data View




Compiled with these configuration settings:

```
iscobol.compiler.gui.window.defaults=gradient-color-1 rgb x#ffffff, gradient-color-2 rgb x#F2F6F9
iscobol.compiler.gui.label.defaults=transparent
iscobol.compiler.gui.tool_bar.defaults=Background-Color rgb x#FFFFFF,Foreground-Color rgb x#F2F6F9
iscobol.compiler.gui.push_button.defaults=flat, Background-color -14675438, Foreground-color x#F2F6F9
iscobol.compiler.gui.entry_field.defaults=border-color rgb x#dae1e5, border-width (0 0 2 0 )
```



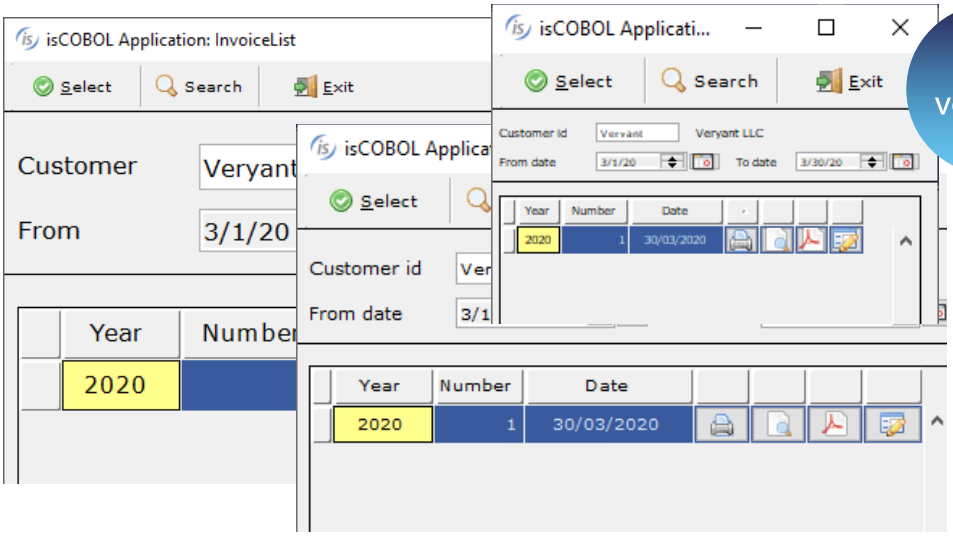

Same Program compiled with above configuration setting

A design language is a definition of a set of design styles and principles for a cohesive look throughout an application or platform. Apple Human Interface guideline, introduced with Apple II in late 1977, Google's Material Design language, developed in 2014 , and Microsoft's Fluent Design System introduced in 2017 are common examples of design languages, and describe the look and behavior your users commonly see every day.

If you want your application to look familiar and up-to-date you will probably end up mimicking one or more of these design languages.

In 2019R2 we've added some features to help you keep your GUI screens fresh – custom borders for entry fields so you can copy Google's input element design with only a bottom border visible, and flat push-button and accordion tab enhancement.

And to make it easy, you can apply these and other styles from the configuration file instead of individually adding to each control's description in each program.




## Zoom Layout Manager

In the last newsletter we talked about the Responsive Layout Manager – letting you apply resizing behavior to each control on your screen to respond to different display environments.

In 2019R2 – we've extended this with the lm-zoom layout manager style. By setting this configuration in your properties file:

```
iscobol.gui.layout_manager=lm-zoom
```

you automatically enable your application to adapt to multiple devices, screen sizes, and screen resolutions. It allows the user to resize an application window or rotate a screen, a required behavior in the era of mobile devices and mobile-first development. Your windows are automatically resizable – the size property of the controls adjust with the window width and the font size property of the controls adjust with the window length.

This provides an easy way to accommodate different screen sizes and resolutions without changing or recompiling your code.

## 2019 R2 HIGHLIGHTS

Here are some of the highlights:

- Insert GUI styles into your program from the configuration file, without changing your code
- We've Improved dynamic variable handling
- We've increased the performance of CALL statements and c-treeRTG operations.
- Our RM/COBOL compatibity has been enhanced to support v8 or greater
- The WebClient console management has become more informative and secure with automaticsession recording and push notifications to the client.

## Compile Error Message Types

**iscc prog1.cbl**
**--S: #36 Ambiguous identifier A; file = prog1.cbl, line = 17, col 11**
**--E: #154 End statement required END-PERFORM; file = prog1.cbl, line = 21, col 19**
**--W: #254 Procedure name same as data name: A; file = prog1.cbl, line = 23**

Do you know what the S, E, and W mean above? When you compile your code, there are types of messages. S is a severe error, so your code won't compile into a class. E is a regular error, and W is a warning – both will not stop the compiler from compiling and creating your .class.

If, for instance, you don't want to see the warning above every time you compile, you can create a configuration property to hide it:
iscobol.compiler.messagelevel.254=0
Read more about this in our documentation.

# JOE - The Power Behind Veryant's Scripting Language

Traditionally software developers have mixed development languages with utilitarian scripting languages to glue their application modules together. These scripting languages (like shell script languages) offer a more relaxed structure, no compilation phase involved, and a flexible and simple way to interact with the operating system that more formal development languages usually lack.

Often legacy applications mix COBOL development with OS-based scripting languages and some key functions of the applications may highly depend on it. If the target development platform lacked the equivalent of that scripting language it would severely complicate the transition. Considering all of that, Veryant offers our own scripting language, called JOE.

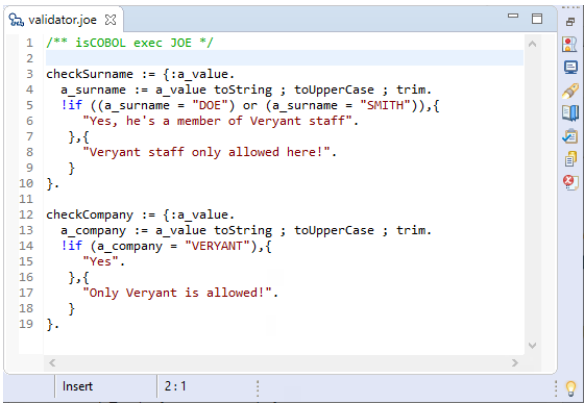
In addition to standard scripting language benefits, JOE can also:

- Interact directly with Java classes
- Interact with isCOBOL classes
- Access and update ISAM files
- Access and update any RDBMS via JDBC driver and SQL
- Access and set environment variables
- Offer basic but powerful logic flow and expression handling
- Interact with Office applications (like Excel) via Apache POI classes or similar
- Work seamlessly with the Application Server Architecture

And as any other scripting language, JOE:

- Is easy to understand, use and extend (beyond the migration phase)
- Can be customized to handle frequent operations
- Resembles any scripting language in terms of capability and readability

Given all of that, JOE is the perfect answer to your migration predicament of what to do with your legacy procedure language. It's the perfect choice for modernization and ongoing development to implement frequent operations with powerful simplicity and elegance, gluing together your application components.




## DATA OPTION

With isCOBOL, you have a wide range of data storage options, which include:

### JISAM

This is the default indexed file format in the isCOBOL runtime framework.

### c-tree

This is another option included with your runtime. Instead of using shared directories for your data files, as JISAM does, c-tree is a file server, adding speed and security to your data. C-tree can be expanded to be a database and allow access to your data with SQL.

### RDBMS with COBOL Syntax

Our Database Bridge product, also known as EasyDB, converts your COBOL File I/O statements to SQL statements and connects with your relational database system

### RDBMS with ESQL

ESQL support is built into the isCOBOL compiler, with no pre-compiling necessary

### File Connectors

Access your data in Vision, c-tree, DBMaker, or Micro Focus ISAM files without converting them.

### DBMaker and Pervasive databases

The DCI and Btrieve file handlers are built into isCOBOL, so you can store your data in these database systems without EasyDB or other additional costs.

Evolution without revolution

## Veryant LLC

**Veryant's dedicated and experienced support team is committed to providing the highest levels of customer care**

For supported customers email us at support@veryant.com

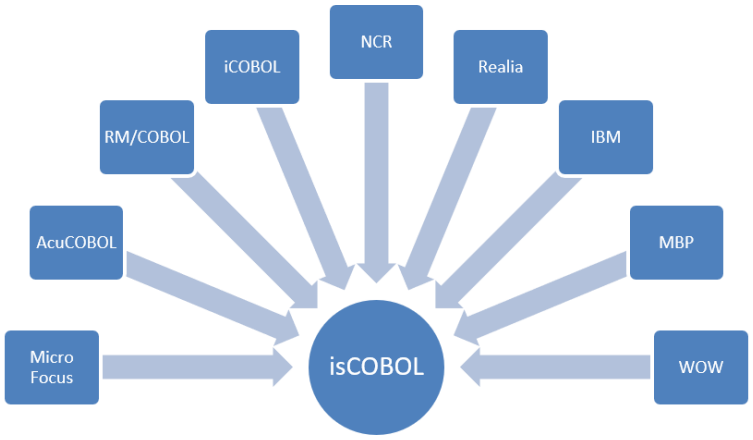
If you would like Veryant to contact you to schedule a technical product briefing, email us at info@veryant.com

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, 2020R1 contains multiple compatibility additions – as we continue to make your conversion process as smooth, quick, and pain-free as possible.**




veryant.com

**Offering choice, flexibility and cost-effective solutions to organizations with valuable COBOL assets**

## CONTACT US FOR A DEMONSTRATION

CLICK HERE

*Cool* Business Oriented Language

Follow Veryant on

veryant.com