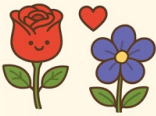




QUARTERLY JOURNAL OF VERYANT AND isCOBOL

Roses are #ff0000,
Violets are #0000ff,
I use hex codes



But I'd use RGB for you.

Get the Right Color in RGB, Hex, or Standard COBOL values using the Color Picker Utility

isCOBOL has a utility called the Color Picker.

You can access it by typing "CPK" in the search bar of your Windows desktop.

It will help you pick the right color every time, and even give you the code you need for many of our GUI controls

Check out the details here:

[Using RGB](#)



The COBOL Technology Innovator
Your Path To Java

veryant

NEWS

THIS ISSUE

1. isCOBOL 2025R2 is here 2. Writing Modern isCOBOL 4. G7 2025- Have you seen this? - Focus on IT 5. Adding File Viewers to WebClient 7. Focus on Search-Panel - Debugging in Thin Client 8. Last page

Big News from Veryant: isCOBOL Evolve 2025R2 Is Here!

We're thrilled to share that the latest version of isCOBOL Evolve—2025R2—is officially available, and it's packed with upgrades that make development smoother, faster, and more modern.

Here's a quick look at what's new:

A Fresh Look for the Debugger: We've added a sleek dark theme and a flat design to the graphical debugger. It's not just easier on the eyes—it's more efficient for developers too.

Smarter IDE Tools: The IDE now includes more toolbar buttons to help you quickly align controls on screens and reports. Plus, our VS Code extension supports workspaces and remote debugging.

Modern UI with FlatLaf: Want your apps to look clean and contemporary? FlatLaf gives you four stylish themes—Light, Dark, Darcula, and IntelliJ. And yes, you can apply them from the command line without changing your code.

Enhanced Controls: Add rollover colors to tree-views, list-boxes, and grids. Tabs on tab-controls now support hints and close buttons, and grids can be loaded dynamically to handle large datasets more efficiently. A new "Today" button makes date-entry control entry quicker and more intuitive.

WebClient Improvements: Now compatible with Java 21, the WebClient starts faster thanks to idle instance preloading. You can launch clusters and sessions with a single command, and pinch gestures make it easier to use on mobile devices.

Ready to explore all the new features?

Upgrade to isCOBOL Evolve 2025R2 today and take your COBOL development to the next level. Visit Veryant's website to learn more and get started

Writing Modern isCOBOL: Moving Beyond Legacy Syntax

Think COBOL is frozen in the 70s with rigid columns and endless STRING statements? Think again. Today's isCOBOL has shed much of its legacy baggage, and you can write code that feels fresh, compact, and even elegant. Here are a few modern features worth using in your next program.

Free Source Format

Forget column restrictions. Write isCOBOL like any other modern language.

Legacy:

- * Comments start with an asterisk symbol in column 7
- * DISPLAY must start at column 12 and end at column 72
DISPLAY 'Hello world: This is my first greeting program.'
'This text goes beyond column 72'.

Modern:

- *> Comments can start anywhere using an asterisk and a greater than symbol
- *> DISPLAY can start in column 1, 2 or any other column you want
- *> and there is no limit at the right side either
DISPLAY 'Hello world: This is my first greeting program. This text goes beyond column 72'.

Cleaner, simpler, no columns to worry about.

Say Goodbye to STRING

Why waste five lines with verbose syntax when one will do?

Legacy:

```
STRING "Hello, " DELIMITED BY SIZE  
    user-name DELIMITED BY SIZE  
    "!" DELIMITED BY SIZE  
    INTO greeting  
END-STRING.
```

Modern:

```
MOVE "Hello, " & user-name & "!" TO greeting
```

Readable at a glance.

Dynamic Memory at Last

Stop hardcoding limits. Use flexible strings and tables.

In the past, you had to declare very large, fixed sizes—like 900 characters or 900 table entries—just in case. Most of the time, much of that space went unused.

Legacy:

```
01 fix-string    PIC X(900).  
01 fix-table    PIC 9(3) OCCURS 900 TIMES.
```

Modern:

```
01 dyn-string    PIC X ANY LENGTH.  
01 dyn-table    PIC 9(3) OCCURS DYNAMIC CAPACITY dt-max.
```

Writing Modern isCOBOL: Moving Beyond Legacy Syntax

Now your code adapts to the real data: dyn-string may hold 10 characters in one case and 9,000 in another, dyn-table may hold 3 numbers at one point and 3,000 later.

In both cases, memory is allocated only for what's actually used.

Loops that Breathe

Break out naturally when it's time.

Legacy:

```
MOVE 0 TO EOF
PERFORM UNTIL EOF = 1
  READ CUSTOMERS NEXT
  AT END MOVE 1 TO EOF
  NOT AT END
  ...
END-READ
END-PERFORM
```

Modern:

```
PERFORM UNTIL EXIT
  READ CUSTOMERS NEXT
  AT END EXIT PERFORM
  NOT AT END
  ...
END-READ
END-PERFORM
```

Cleaner than cramming logic into a single condition.

Built-in Functions > External Calls

Built-in functions can replace external CALLs. For example, uppercasing a string:

Legacy:

```
CALL "TOUPPER"
  USING name, upper-name.
```

Modern:

```
MOVE $UPPER-CASE(name) TO upper-name.
```

One simple line instead of writing and calling an external subprogram.

What's Next?

Modern COBOL is concise, flexible, and developer-friendly. And this is just the start. Next issue, we'll explore **Object-Oriented isCOBOL**: yes, classes and methods, in isCOBOL.

FOCUS ON IT

Put program focus on the search panel

If you want to give the focus to the search panel of a grid you can use the following statement:

modify grid1 action action-search

This statement is enough if the focus is on the grid. If the focus is on another control you need to force the focus to the grid with the following statements

move grid-id to control-id

move 4 to accept-control

G7 in Rome!

In today's globalized market, expanding into diverse regions with varying cultures, needs, and work practices requires more than ambition—it demands strong local partnerships. Before seeking clients, it's essential to identify reliable partners who share your goals, strategic vision, and technical expertise. These alliances open doors to markets that would otherwise be inaccessible.

In 2019, we joined forces with other companies to launch a collaborative initiative aimed at delivering technical and economic benefits for all involved. This led to the creation of the annual "Strategic Partner Conference," hosted in rotation by each participating country. The event focuses on current topics in computer science and provides a platform to share experiences and insights from the past year.

For 2025, Veryant Italia hosted the fourth edition in Rome, welcoming companies from Brazil, the Philippines, Japan, Thailand, Taiwan, the USA, and Italy. The conference in mid-October spotlighted two key themes: Artificial Intelligence in programming languages and Digital Transformation.



Have You Seen This?

Newest Video:

Keep up with new features in ["2025 Release 1 Introduction"](#).

Learn how to connect to external vendors with REST in ["Consuming a REST Service"](#)

New KB Articles:

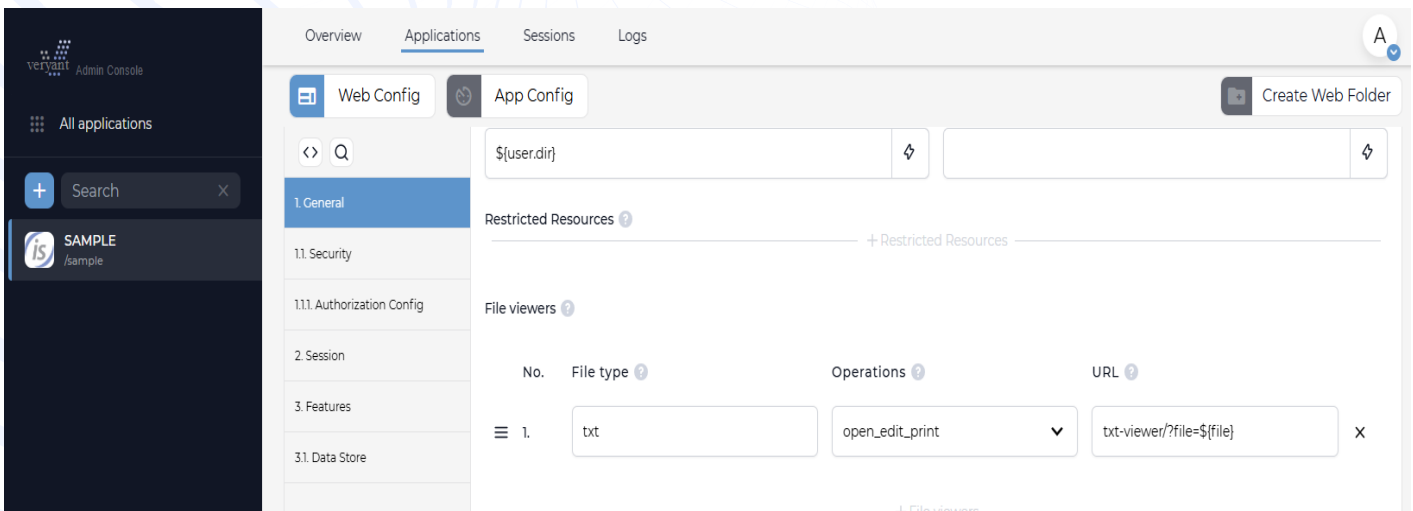
[How to validate an XML document](#)

[How to easily change the font of a character program](#)

Adding File Viewers to WebClient

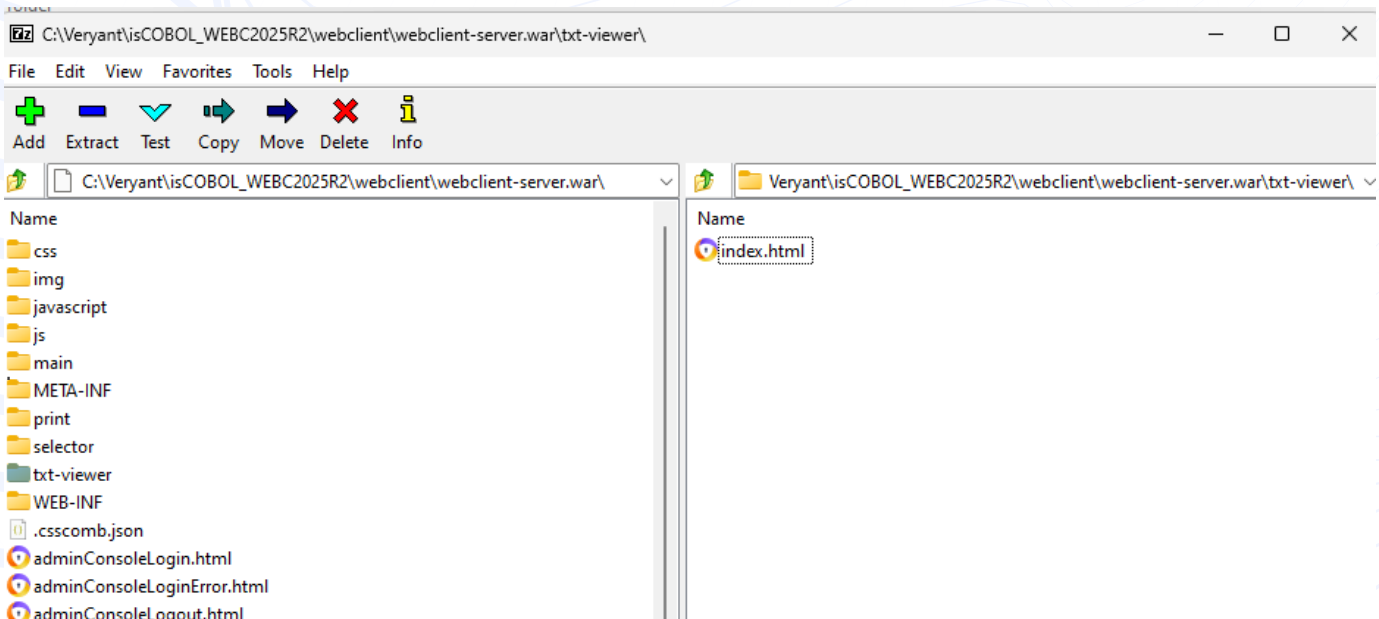
COBOL applications can generate output files (such as .txt or .pdf) and use the C\$DESKTOP routine to open or print these files with their associated applications.

In WebClient's "File Viewers" section, you can configure a mapping between file extensions and URL handlers to manage specific AWT Desktop file operations



In the configuration shown above, files with a .txt extension are processed using the /txt-viewer URL handler when CDESKTOP-OPEN, CDESKTOP-EDIT or CDESKTOP-PRINT is invoked on the file.

To implement a URL handler, create a folder in the webclient-server.war directory and include an index.html file inside it.



Adding File Viewers to WebClient

Here's an example of a URL .txt handler

```
<!DOCTYPE html>
<html>
  <head>
    <title>Demo text file viewer</title>
    <meta charset="utf-8">
    <script language="javascript" type="text/javascript">
      const searchParams = new URLSearchParams(window.location.search);
      const blobId = searchParams.get('file');
      console.log("blobId", blobId)

      document.addEventListener("DOMContentLoaded", function(event) {
        var fileContent = document.getElementById('filecontent');
        fileContent.value = 'Loading file content ...';

        fetch(blobId)
          .then(response => response.blob())
          .then(blob => {
            const reader = new FileReader();
            reader.readAsText(blob);
            reader.onloadend = function() {
              fileContent.value = reader.result;
            };
          })
          .catch(error => {
            fileContent.value = "Blob error " + error
            console.error('blob error', error)
          });
      });
    </script>
  </head>
  <body>
    <h1>Text file viewer</h1>
    <textarea id="filecontent" rows="20" cols="110" style="white-space: pre;font-family:monospace;font-size: 12px" readonly></textarea>
  </body>
</html>
```

New Easier Search in the current isCOBOL Documentation



Since version 2025 R1 the isCOBOL HTML documentation includes an improved search feature?

Now you can search for substrings by using wildcards. This is particularly useful when you're looking for a configuration property. In previous documentations you had to specify this prefix and know the first letters of the property you're looking for. For example, to find "iscobol.code_prefix" you had to type at least "iscobol.cod" in the search box. Now instead you can use wildcards and type "*code*pref*".

Search results are returned according to the searched word or substring weight. Pages where the searched word or substring appears more times or where it appears in a title rather than in the paragraphs' text are returned first.

Good Practice for Debugging in Thin Client



The programs you run in a live environment should always be compiled in release-mode, not debug-mode. But what if there are several active users connected to the isCOBOL Server and one of the users complains about an odd behavior that you want to debug one or more programs to verify?

In this scenario, you might think it would be best to copy debug-compiled classes to the live environment's program folder, overwriting the existing release-compiled classes in order to run a client in debug mode. You would then need to restore the release-compiled classes when you're done with your debugging.

But this approach has a very important side effect: **classes compiled in debug-mode are not fully thread safe and may generate memory leaks** when they're loaded by clients that are not running in debug mode and use the isCOBOL Server's main JVM.

This can cause serious problems, especially if the isCOBOL Server is rarely restarted.

A better practice for debugging in a thin client environment is by isolating the debug-compiled classes either in a separate code-prefix or in a separate isCOBOL Server. The documentation explains in detail how to do this in [isCOBOL Application Server : Good practice for the use of programs compiled in debug mode.](#)

PLEASE JOIN US ON

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



Watch our demonstration videos and Subscribe to our YouTube Channel



Vampire



Sunlight



Superman



Kryptonite



Programmers

Documentation



Evolution, without revolution



Contact Us

For supported customer email us at support@veryant.com

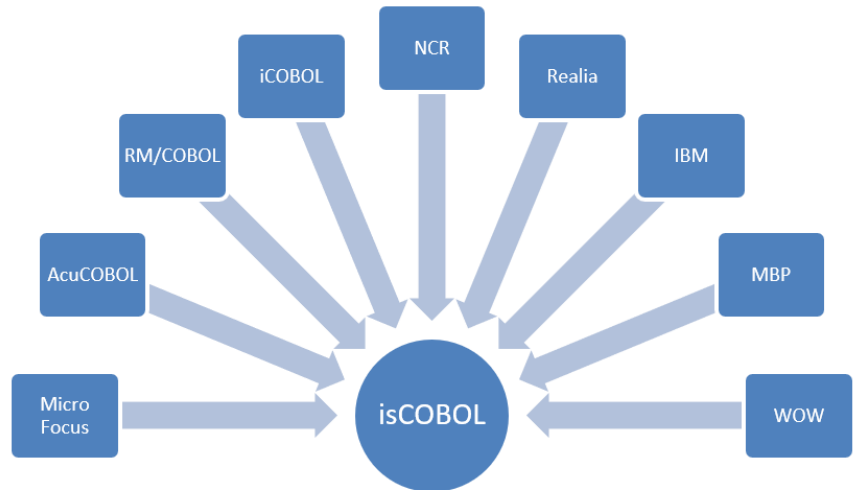
If you would like Veryant to contact you to schedule a technical product briefing, email us at info@veryant.com

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

©2025 Veryant - All Rights Reserved