isCOBOL™ Evolve

isCOBOL Evolve 2022 Release 1 Overview

isCOBOL Evolve 2022 Release 1 Overview

Introduction

Veryant is pleased to announce the latest release of isCOBOL™ Evolve, isCOBOL Evolve 2022 R1.

isCOBOL Evolve provides a complete environment for the development, deployment, maintenance, and modernization of COBOL applications.

The new version now supports all Java LTS releases: Java 1.8, Java 11 and Java 17.

isCOBOL IDE is now based on a newer Eclipse version, 2021-09 (4.21).

WebClient now supports balancing through clustering and a restyled administration UI.

2022R1 also includes many GUI improvements and new compatibility features to simplify the migration of existing COBOL compiler to isCOBOL Evolve suite.

Details on these enhancements and updates are included below.

isCOBOL Runtime

isCOBOL Evolve 2022R1 is certified for Java 17, the current Java LTS (Long Term Support) release. It is important to have isCOBOL Evolve 2022R1 qualified to be used by the latest LTS version of Java because LTS applies the tenets of reliability engineering to the software development process and software release life cycle. Long-term support extends the period of software maintenance; it also alters the type and frequency of software updates (patches) to reduce the risk, expense, and disruption of software deployment, while promoting the dependability of the software. It does not necessarily imply technical support. In addition, starting from Java 17, Oracle JDK 17 and future JDK releases are provided under a free-to-use license until a full year after the next LTS release.

isCOBOL Evolve 2022R1 now supports all Java LTS releases: Java 1.8, Java 11 and Java 17. WebClient currently does not support Java 17, but we plan to have it ported in the 2022R2 release.

In addition to Java 17, isCOBOL Evolve 2022R1 is also fully qualified to be used under Microsoft Windows 11 in combination with Java 11.0.13 (see https://www.oracle.com/java/technologies/javase/products-doc-jdk11certconfig.html)

New library routines and configuration settings have been added to allow multiple configuration files to be used when running isCOBOL applications

New configurations

New configuration settings have been added in this release:

- **iscobol.conf.copy=FilePath** to include a separate configuration file in the current configuration file.

  This setting can be used to easily maintain and use multiple configuration files. For example, when running the following command:

  ```
  iscrun –c myapp.properties MAINPROG
  ```

  and the myapp.properties file contains:

  ```
  iscobol.conf.copy=default.properties
  ...
  iscobol.conf.copy=customer-specific.properties
  ```

the program is run using the settings specified in the `myapp.properties` configuration file, along with settings from the `default.properties` and `customer-specific.properties` files. If the same configuration property name is set in multiple configuration files, the last defined one takes precedence over the others, thus overwriting previous values.

- **`iscobol.file.index.fileversion=n`** to create c-tree files that are backward compatible. For example, you can set the value to 2 to make the current C-Tree RTG v3 create files compatible with the previous C-Tree RTG v2.
- **`iscobol.file.output_lock=false`** to prevent an exclusive lock during open output. The default value is **`True,`** so that an OPEN OUTPUT is treated as OPEN OUTPUT WITH LOCK.
- **`iscobol.file.extend_lock=false`** to prevent an exclusive lock during open extend. The default value is True, so that an OPEN EXTEND is treated as OPEN EXTEND WITH LOCK.

  The previous two configurations can be set in rare cases of lock needs. By default, they are set to true for higher performance of WRITE operations after the file is opened.
- **`iscobol.display_message_timeout=n`** to specify a timeout in hundreds of a second for error message boxes. When the timeout expires, the error message box is automatically closed as if the user pressed the OK button. When set in the runtime configuration it affects every Java exception that is usually reported via graphical message box. When set in the isCOBOL Client local configuration it affects connection error and session termination messages.

Additionally, the existing configuration `iscobol.file.page_eject_on_close=true` is now supported on files assigned to -P SPOOLER-DIRECT as well as other print files.

New library routines

Two new library routines have been implemented:

- C$CONFIG to reset, load or append a new configuration properties file.

  This library routine can be used to reset the runtime configuration by setting a new configuration file, or to append a configuration file to the current configuration after the main program is started. It can be useful, for example, to load a different configuration file after logging in. The following snippet:

```
call "c$config" using cconfig-reset,
                      "conf2.properties"
```

  resets the currently loaded configuration and loads a new configuration contained in the conf2.properties file.

  This code snippet:

```
call "c$config" using cconfig-append,
                      "conf3.properties"
```

  appends settings contained in the “conf3.properties” file to the current configuration.

- W$GETC is used to retrieve the keystroke pressed. This routine is useful when the program needs to perform a low-level intercept of the key pressed in a character-based user interface.

  The following code snippet shows how to intercept the F1 key:

```
77  wcharacter              pic xx.
...
    call "w$getc" using wcharacter
    if wcharacter = "k1"
       display "pressed F1"
```

These routines can be used in new programs, and also increase compatibility with other COBOL dialects, such as ACUCOBOL-GT.

A new op-code named WINPRINT-GET-NO-ASYNC-JOBS has been added to the WIN$PRINTER routine to inquire how many async print jobs are currently running. This is useful when using asynchronous print jobs to check if some are still running. For example, this code snippet checks if the async jobs previously executed are terminated:

```
call "win$printer"  using WINPRINT-GET-NO-ASYNC-JOBS, n-jobs
                    giving winprint-status
if n-jobs = 0
  ...
```

isCOBOL IDE enhancements

The isCOBOL IDE 2022 R1, now based on Eclipse 2021-09 (4.21), requires at least Java version 11 by default, but offers full support for Java 17. There are many improvements in performance stability and usability on a day-by-day usage. Below is an extract of the new functionalities:

- A new "System" theme is available in the Appearance preference page. This theme is built using system colors, and as a consequence integrates well with any OS and OS theme. Also, there were many enhancements on “dark” theme to style better menus, drop down controls, tool-tips, progress bars and more.
- A new preference, Enable word wrap, is available in the Console preference page. This setting persists the current state of the "Word wrap" toggle onto the console view between user sessions. By default, word wrapping is disabled on console output.
- In the Console view, you can repeat your last search in the forward or backward direction in the following ways: Right-click in the Console view, then select Find Next or Find Previous.
- Previous Edit Location navigation (formerly named Last Edit Location) is now expanded to remember multiple edit locations. The last 15 edit locations are now remembered. For convenience, similar edit locations in close proximity to each other are also merged so that each of the 15 remembered locations remains distinct. Ctrl+Alt+LEFT_ARROW (or on Mac Ctrl+Opt+LEFT_ARROW) navigates to the most recent edit location, just as Ctrl+Q always has in prior releases.
- You can now scroll horizontally in the Text Editor using Shift+Mouse Wheel and touchpad gestures on Windows. Horizontal scrolling with touchpad already works on Linux and MacOS.

WebClient

isCOBOL WebClient, Veryant's solution for running desktop applications in a browser, has a more modern and better organized administration console in the 2022R1 release that makes configuration easier, and includes a new load balancing solution.

New administration user interface

The admin user interface has received an overhaul, and is now more structured in how all parameters are presented to the administrator. It provides categories that logically group parameters, making it much easier to navigate settings, both for server configuration and application configuration.

A server status page lets you keep track of the number of session pools available to applications, the number of currently active users, the number of running sessions, how many connections are established, and the number of configured and enabled applications.

The sessions' view allows administrators to monitor currently running applications, and, if allowed, to view and take control of users' sessions.

Figures 1, 2 and 3 depict the new administration panel, and the new grouping of the configuration items in categories.

Figure 1 – *Server configuration page*

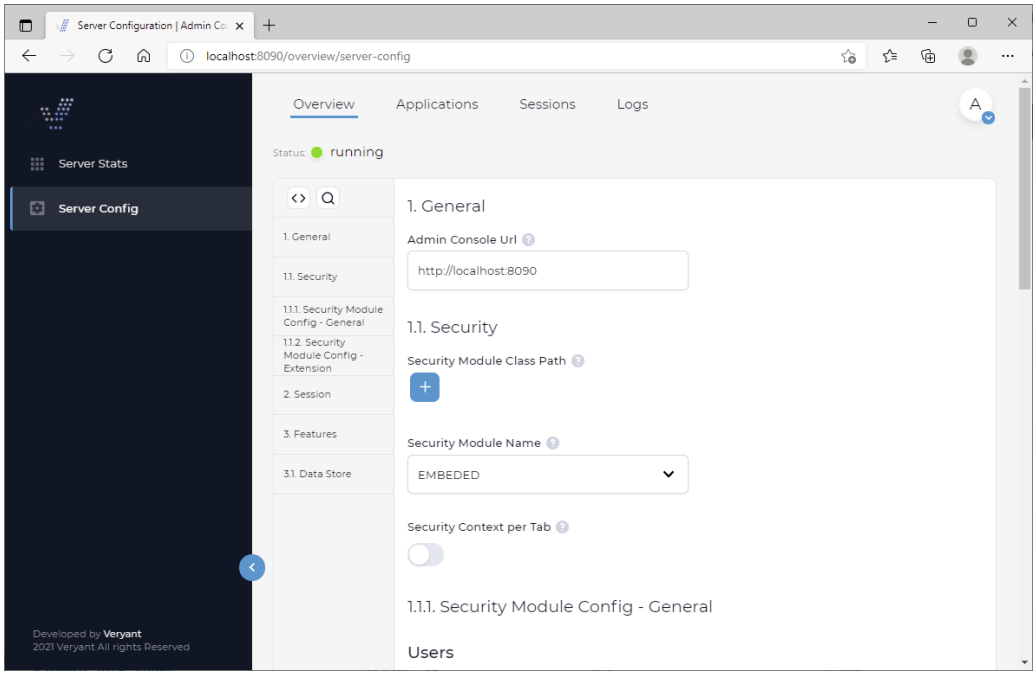


Figure 2 – *Web application configuration*

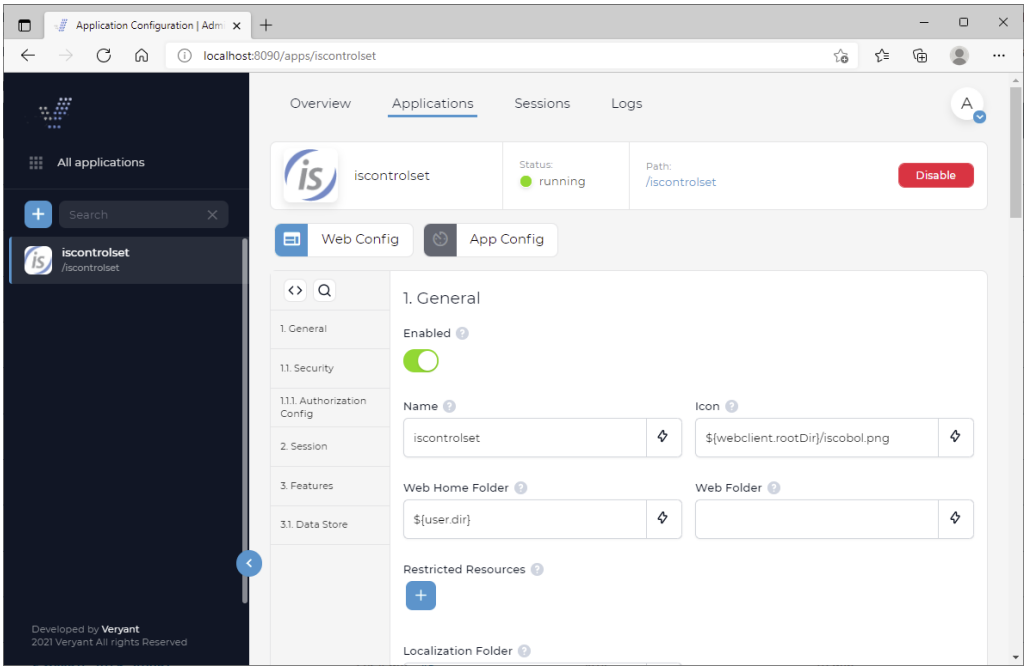
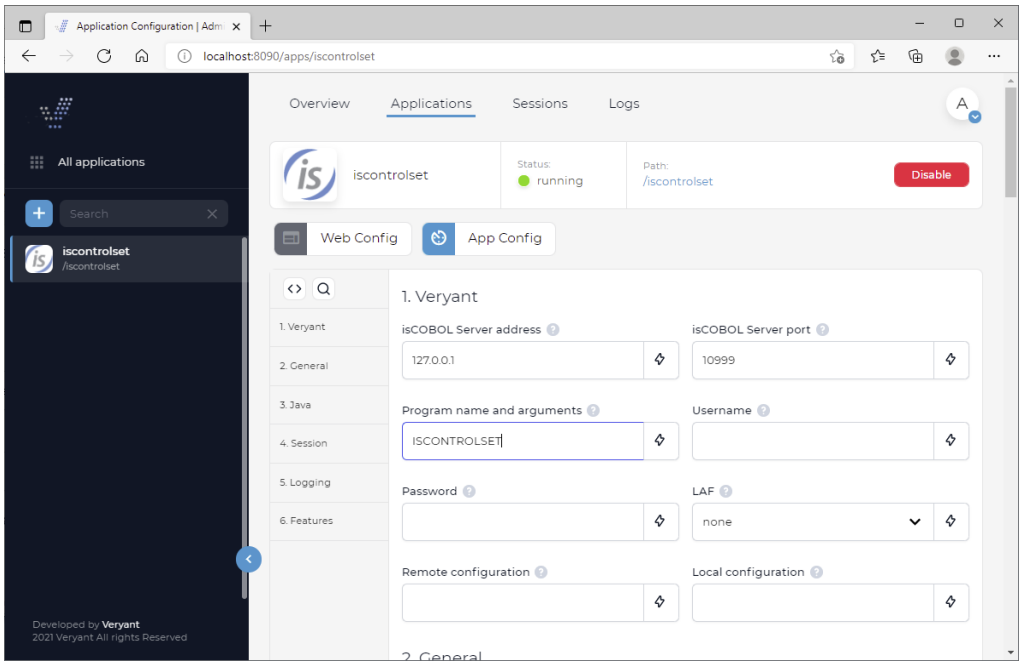


Figure 3 – *Application configuration*



WebClient Balancer

WebClient now supports balancing through clustering; separating WebClient into modules that can be divided into multiple instances on multiple servers to enable you to handle larger loads. These modules are

- WebClient Cluster Server which handles web requests
- Application pool which handles running COBOL applications
- Admin Console which handles the server administration pages.

A minimal installation requires 1 cluster server and 1 session pool, and more cluster servers can be added for redundancy. As a general rule, no more than 2 cluster servers should ever be needed in most cases, as it is a very lightweight task that basically resends messages from the application instance and the browser. It also picks an application pool in which to run the requested application.

Session pools can be scaled dynamically, based on the number of connected users or resource usage.

Cluster Server is a stateless web server, which means that if you have multiple Cluster Servers it doesn't matter which server you connect to from the browser, even on refresh, the browser connection will always find the way to its application instance. If you have multiple Cluster Servers deployed and one of the Cluster Servers terminates, all the browser connections to this server can be reconnected through the other running Cluster Server without terminating the application instance.

Inside Cluster Server is a built-in mechanism - a Session Pool load balancer. This mechanism is responsible for finding a free Session Pool where a new instance will be started. The load balancer uses a round-robin algorithm.

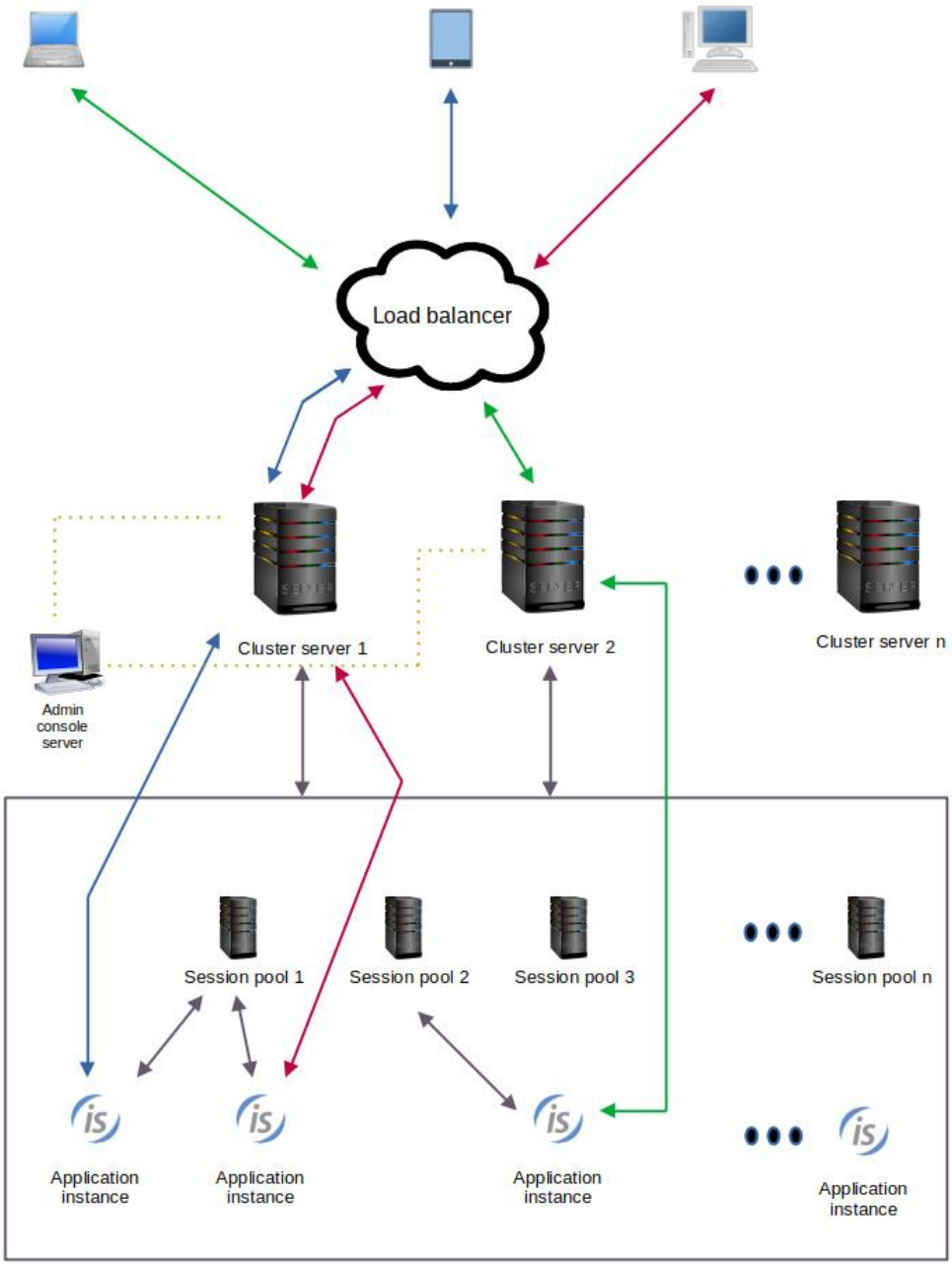
Everything in your application that is related to web is configured and provided through the Cluster Server. This includes security (SSL, authentication, logout), translations, fonts or custom web resource handling. On the other hand, everything related to your application binary is configured and handled in Session Pool. Because of this the webclient.config file needs to be divided into 2 separate configuration files - webclient-server.config which is used by Cluster Server and web client-app.config which is used by Session Pool.

A single Admin console can be used to handle configuration of all Cluster Servers by configuring the list of websocket URLs of all Cluster Servers in the Admin Console configuration file.

Modules inside a Cluster deployment are connected through WebSocket. To make sure that no other malicious servers connect to your Cluster Servers, all modules must share the same secret key which is used in handshake when establishing the websocket communication. The secret key is defined in webclient.properties, webclient-sessionpool.properties and webclient-admin.properties as property webclient.connection.secret. Please change this secret key to your random value before going into production. This secret key is also used for signing JWT tokens that are used for user session cookies. Make sure you use only alphanumeric characters in the secret key, and do not use any special characters.

Figure 4 – Communication flow describes the flow of communication between devices running the application. A device connects to the load balancer, which connects to a Cluster Server. The server will run an application using one of the available session pools.

Figure 4 – *Communication flow*



GUI enhancements

Many improvements to GUI controls have been implemented in this release. Library routines W$BITMAP and W$MENU now allow merging different symbol fonts and to set new attributes in the Hamburger menu. New configuration settings can be used to modify the look and feel of modal windows and to have finer control with the LM_ZOOM layout manager.

GUI controls

A new property named MARGIN-WIDTH is supported in Entry-Field controls to set or retrieve the spacing between text and borders. The property is a list of four values that specify the padding space in pixels from the top, left, bottom and right border respectively (the same rule of the existing BORDER-WIDTH property). When used on single line entry-fields, only the left and right borders are meaningful, since text is always vertically centered.

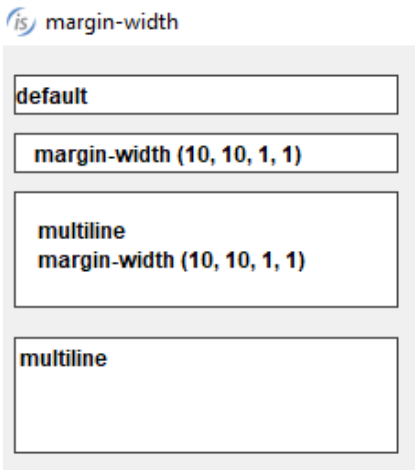
The same spacing applies to the PLACEHOLDER property, if defined. This feature, known also as Inset, helps in defining different spacing of different controls. Alternatively, you can use a different margin space in all entry-field controls of the entire isCOBOL application by using the compiler code injection feature.

For example, the entry-fields defined below use the new syntax in the second and third entry-fields.:

```
03 entry-field
   line 2  col 2 size 30 cells.
03 entry-field
   line 4  col 2 size 30 cells
   margin-width (10, 10, 1, 1).
03 entry-field multiline line 6 col 2 lines 4 cells size 30 cells
   margin-width (10, 10, 1, 1).
03 entry-field multiline line 11 col 2 lines 4 cells size 30 cells.
```

The result when running is shown in in Figure 5, *Entry-field MARGIN-WIDTH property*.

Figure 5. Entry-field MARGIN-WIDTH property.



The FLAT style is now supported in Tree-View controls to have more modern-looking buttons when running with the Windows LookAndFeel. For example, having this code in the screen section declaration:

```
01  Mask.
    03 Tv1 tree-view buttons lines-at-root
       line 2 col 2 lines 17 size 40 cells
       show-sel-always selection-background-color rgb x#6883AE
                       selection-foreground-color rgb x#FFFFFF.
    03 Tv1 tree-view buttons lines-at-root FLAT
       line 2 col 44 lines 17 size 40 cells
       show-sel-always selection-background-color rgb x#6883AE
                       selection-foreground-color rgb x#FFFFFF.
```

with the FLAT style is set only in the second tree-view, will result in the running program looking as shown in Figure 6, *Tree-view FLAT style*, where you can notice the difference in the styling of the two tree view controls.

Figure 6. Tree-view FLAT style.

The TRANSPARENT-COLOR property, already supported with BITMAP controls, is now supported on PUSH-BUTTON controls. A specific color can be specified to represent the transparent color in the button's bitmap image, even if the bitmap has no transparency.

The layout-manager syntax now supports "max-font-zoom=n min-font-zoom=n" clauses to set the maximum and minimum zoom levels that can be applied during window resizing when used in conjunction with the LM-ZOOM layout manager. The values are expressed as a percentage in which the font will be altered. When the zoom level is outside this range, LM-ZOOM behaves exactly as LM-SCALE, resulting in changes only to the dimensions and coordinates of the controls without affecting the font size. For example, the code snippet:

```
77  my-lm-zoom handle of layout-manager LM-ZOOM
       "max-font-zoom=150 min-font-zoom=50".
```

declares a layout-manager for LM-ZOOM allowing font changes from 50% when decreasing and 150% when increasing.

A new event named MSG-FINISH-FILTER is fired in grid controls when the visible rows in the grids change because of changes applied using the Filters or Search panel features. This is useful if the program needs to be aware of a filter being applied to the grid's content.

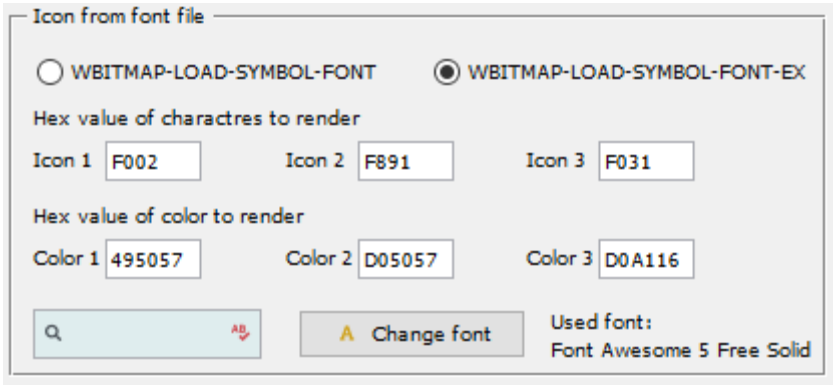
Library routines

A new op-code named WBITMAP-LOAD-SYMBOL-FONT-EX in W$BITMAP library has been implemented to load icons and to merge different font symbols or font styles. This lets you create a handle with a strip of icons loaded from different font handles, or from the same handle but with different styles, such as the color. The code snippet:

```
initialize wbitmap-lsf-data
move h-font        to wbitmap-lsf-font(1)
move character-1-n to wbitmap-lsf-characters(1)
move color-1-hex   to wrk-color-hex
perform CALCULATE-COLOR
move wrk-color     to wbitmap-lsf-color(1)
move h-font        to wbitmap-lsf-font(2)
move character-2-n to wbitmap-lsf-characters(2)
move color-2-hex   to wrk-color-hex
perform CALCULATE-COLOR
move wrk-color     to wbitmap-lsf-color(2)
move h-font        to wbitmap-lsf-font(3)
move character-3-n to wbitmap-lsf-characters(3)
move color-3-hex   to wrk-color-hex
perform CALCULATE-COLOR
move wrk-color  to wbitmap-lsf-color(3)
call "W$BITMAP" using wbitmap-load-symbol-font-ex,
                      16, wbitmap-lsf-data
giving h-font-icon
```

creates the handle h-font-icon with 3 icons from different characters of the same font handle (h-font in this case loaded from Font Awesome 5 Free Solid) but using 3 different colors, and the 3 images can be used with the standard bitmap-handle and bitmap-number syntax as shown in the entry-field and push-button in Figure 7, WBITMAP-LOAD-SYMBOL-FONT-EX new op-code.

Figure 7. *WBITMAP-LOAD-SYMBO-FONT-EX new op-code*



New attributes for the W$MENU library have been added to customize the hamburger menu layout and behavior. This is the list of new attributes:

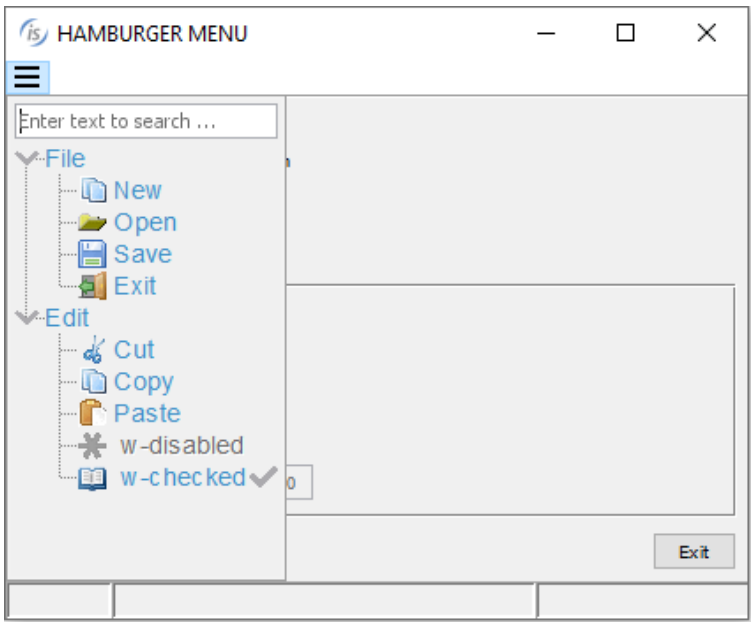
- `EXPANDED,` to specify if the menu items with children should be automatically expanded
- `HEIGHT,` to set the height in pixels of the area covered by the hamburger menu
- `LAYOUT-MANAGER,` to set the rules of layout-manager bound to the menu that will applied when the window is resized while the menu is open
- `SEARCH-PANEL,` to enable the tree view's search panel, allowing the user to filter the content
- `STATUS-BAR-COVERING,` to prevent the hamburger menu's tree view to overlap the window's status bar.

For example, this code snippet sets three of the new attributes:

```
call "w$menu" using wmenu-set-attribute, "expanded", "yes"
call "w$menu" using wmenu-set-attribute, "search-panel", "yes"
call "w$menu" using wmenu-set-attribute, "status-bar-covering", "no"
```

and in Figure 8, *Hamburger menu new attributes*, you can see the new look of the hamburger menu when opened.

Figure 8. Hamburger menu new attributes.



The W$CREATEFONT library routine can now load fonts that have been embedded in the class using the COPY RESOURCE directive. This allows you to use a font without needing it to be installed on the client PC. For example, this code snippet:

```
copy resource "../resources/Font Awesome 5 Free-Solid-900.otf".
call "w$createfont" using "Font Awesome 5 Free-Solid-900.otf"
                          "Font Awesome 5 Free Solid".
```

uses the COPY RESOURCE directive to include that .otf file in the class at compile time, and the W$CREATEFONT to load it.

Configuration settings

New configuration settings have been implemented:

iscobol.gui.windows_darkening=n to specify the opacity of the parent window when modal floating windows and message boxes are opened. A positive value ranging from 1 to 100 specifies the level of darkening with 1 representing an imperceptible darkening and 100 representing the black color, while a negative value ranging from -1 to -100 specifies the level of transparency with -1 representing an imperceptible transparency and -100 representing full transparency.

This is especially useful with applications running in WebClient, to emulate the effect of modern web applications when modal windows are opened. Thin Client applications can also take advantage of this property for a more modern look and feel of the application.

For example, configuring: `iscobol.gui.windows_darkening=15`

the window shown in Figure 9, *Window before darkening effect*, has its normal color during the accept, but when the message box appears, the color is automatically changed and shown in in Figure 10, *Window after darkening effect*.

Figure 9. Window before darkening effect.

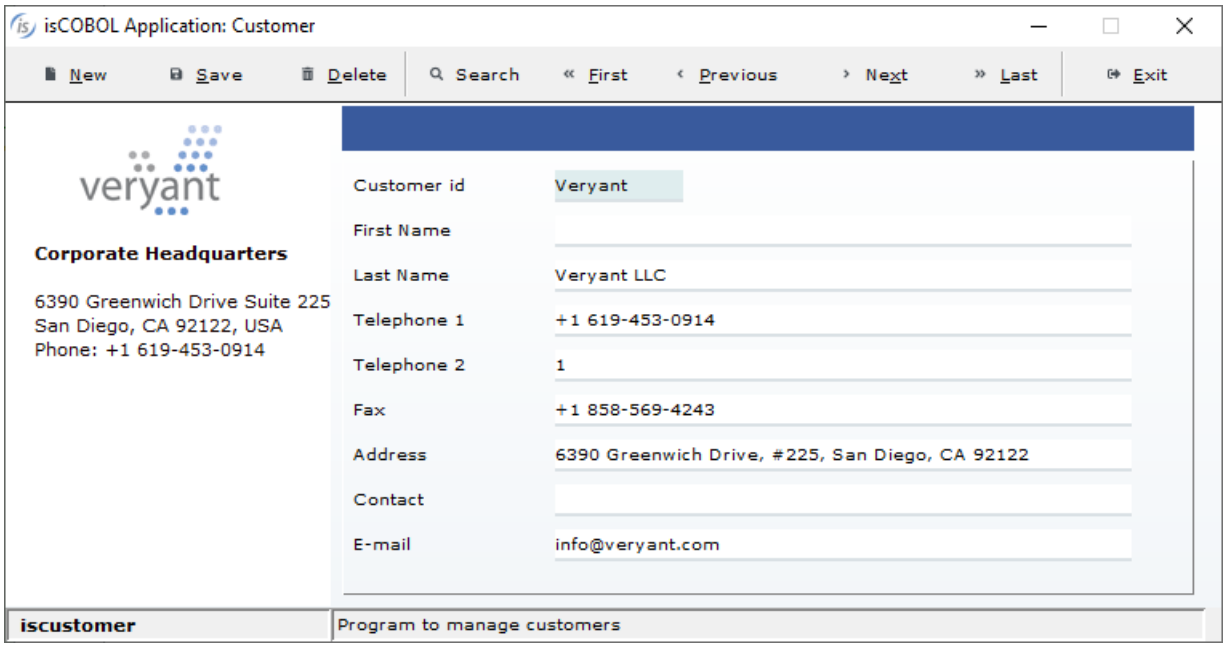
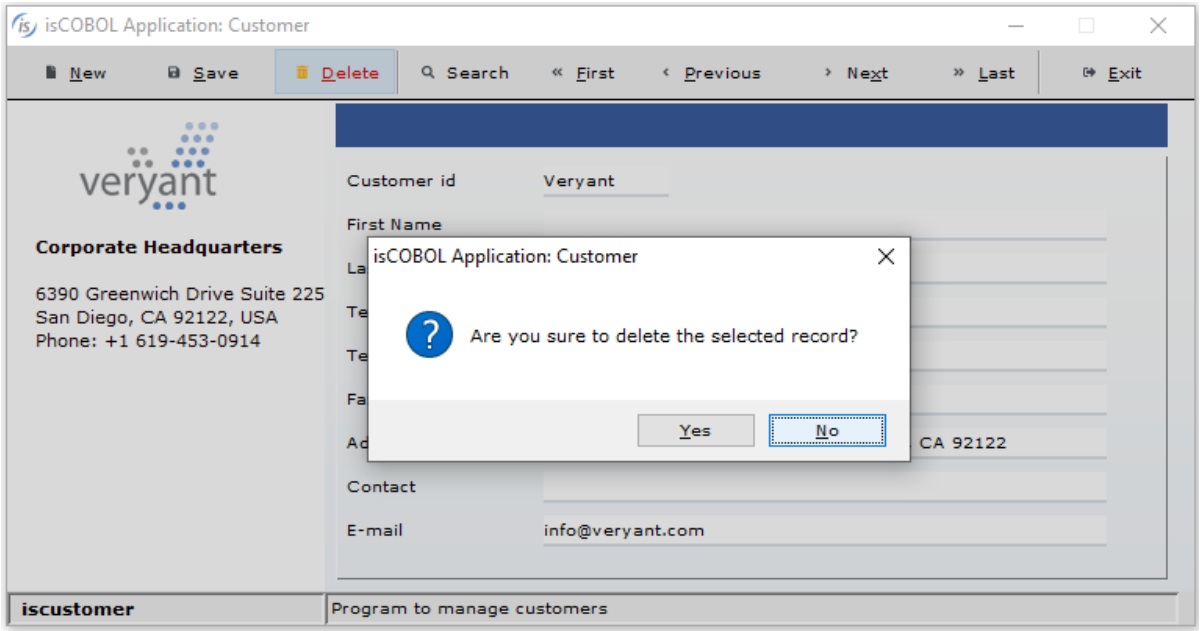


Figure 10. Window after darkening effect.



iscobol.gui.layout_manager.max_font_zoom=n to specify the upper limit for the font increase performed by the LM-ZOOM layout manager

iscobol.gui.layout_manager.min_font_zoom=n to specify the lower limit for the font reduction performed by the LM-ZOOM layout manager

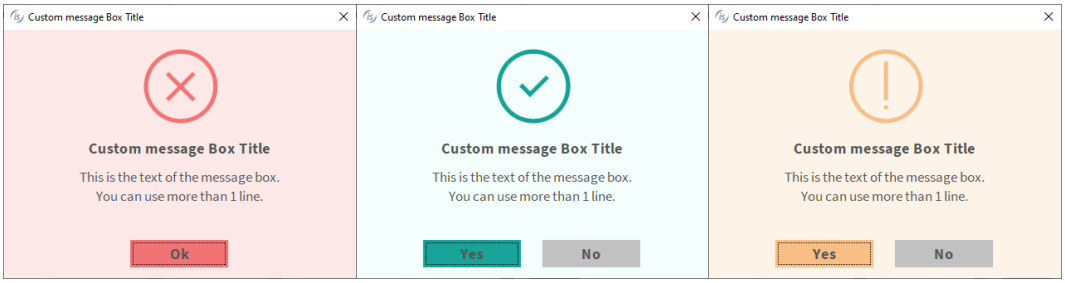
iscobol.gui.light_gray_is_transparent=false to avoid the transparent effect to be applied automatically on color 0xc0c0c0

Additionally, in the existing configuration `iscobol.gui.messagebox.custom_prog`, the value of the program name can be followed by comma and a letter to specify the program location. When 'C' is used, the program is searched for in the client machine, otherwise it's searched for in the server machine. For example, configuring:

```
iscobol.gui.messagebox.custom_prog=CUSTOMMSG,S
```

the Cobol program named CUSTOMMSG is called when programs display a MESSAGE BOX, and this program is searched for on the Server. Figure 11, *An example of a custom message box*, shows some message box fully customized.

Figure 11. An example of a custom message box



isCOBOL Compiler

The isCOBOL 2022R1 compiler includes improved compatibility with other COBOL dialects to simplify migrations, and isCOBOL ESQL has been improved as well to ease migration from the IBM DB2 PRE (ESQL precompiler).

Improved Compatibility with other COBOLs

Enhancements have been made in this release to improve our compatibility with MicroFocus COBOL, such as:

- `WITH DEBUGGING MODE` in `SOURCE-COMPUTER` is now supported to handle code lines marked with D. For example, this code snippet:

```
    CONFIGURATION SECTION.
    SOURCE-COMPUTER.  WITH DEBUGGING MODE.
    working-storage section.
    ...
    procedure division.
    MAIN.
D       DISPLAY "LOG: calling ELCUST"
        CALL "ELCUST" USING PARAMS
D       DISPLAY "LOG: returned from ELCUST"
```

turns on debugging mode, causing the lines containing the DISPLAY statements to be included in the compiled program. If the WITH DEBUGGING MODE statement is commented out, like this:

```
    SOURCE-COMPUTER. . *>WITH DEBUGGING MODE.
```

all lines marked with D in the Indicator area, column 7 for ANSI sources, are treated as comments and the DISPLAY statements will not be included in the compiled program.

- the XML syntax extension with ORGANIZATION IS XML in SELECT and XD for record definition is supported to clean compile.

A code such as:

```
FILE-CONTROL.
select xml-customer assign "out-customer.xml"
      organization  is xml
      document-type is "customer"
      file status is xml-status.
file section.
xd xml-customer.
01  rec-customer identified by "customer".
    05  xmls-name pic x(80) identified by "elementName" is attribute.
    05  xmls-age  pic 9(18) identified by "elementAge".
...
    open output xml-customer
    write rec-customer
    close xml-customer
```

is now compiled without needing compiler options, enabling reading and writing XML files.

- the direct syntax is now supported on INVOKE statement, using the code snippet:

```
invoke JSystem::"getProperty"("user.home") returning w-home
```

or:

```
invoke JSystem:>getProperty("user.home") returning w-home
```

are equivalent of:

```
invoke JSystem "getProperty" using "user.home" returning w-home
```

Enhancements have been made to improve the compatibility with RM/COBOL, such as:

- a new compiler option, -crko, has been implemented to declare keys of indexed files in offset order including segments. This option is useful to have the same file definitions of indexed files, both when isCOBOL is used to access existing RM indexed files through RMC (RM file Connector) or when using the ISMIGRATE utility to migrate the existing files to the desired Veryant file system such as C-Tree or JISAM.

- enhanced the existing -cr option to better manage the character DISPLAY with POS declared without LINE.

An enhancement in the compiler has been implemented to improve the compatibility with ACUCOBOL-GT:

- the COPY RESOURCE directive is now fully supported. It allows you to embed any file in the class. This is typically useful to load an image using the W$BITMAP routine that is not present on the disk and needs to be embedded in the program class. Code like the following:

```
copy resource "..\resources\logo.png".
call "w$bitmap" using wbitmap-load, "logo.png"
                giving h-bmp.
```

will load the image “logo.png” and embed it in the class at compile time.

Improved Compatibility with DB2 ESQL Precompiler

The *Rowset positioning* syntax is now supported by ESQL cursors, to simplify the migration from DB2 Precompiler to the isCOBOL compiler, that supports ESQL statements accessing directly with the JDBC driver. This feature enables programs to read one or more records from a cursor starting either from the beginning of the cursor or from a given position in the cursor. As an example, given the following table:

```
CUST_ID CUST_NAME
------- ---------
1 Daniel
2 Robert
3 Bill
4 John
5 Richard
6 Eleonore
7 Bob
8 Cindy
9 Rachel
```

Create a cursor with the following syntax:

```
exec sql
  declare cur cursor
     with rowset positioning
     for select * from customers
end-exec
```

After opening the cursor, you can read record 4 (John) and 5 (Richard) with the following syntax:

```
exec sql
   fetch rowset
      starting at absolute 3
      cur
      for 2 rows
      into :wrk-id, :wrk-name
end-exec
```

You can then read records 7 (Bob), 8 (Cindy) and 9 (Rachel) with the following syntax:

```
exec sql
   fetch rowset
      starting at relative 1
      cur
      for 3 rows
      into :wrk-id, :wrk-name
end-exec
```

The ABSOLUTE clause counts records from the beginning of the cursor while RELATIVE counts from the current position in the cursor.

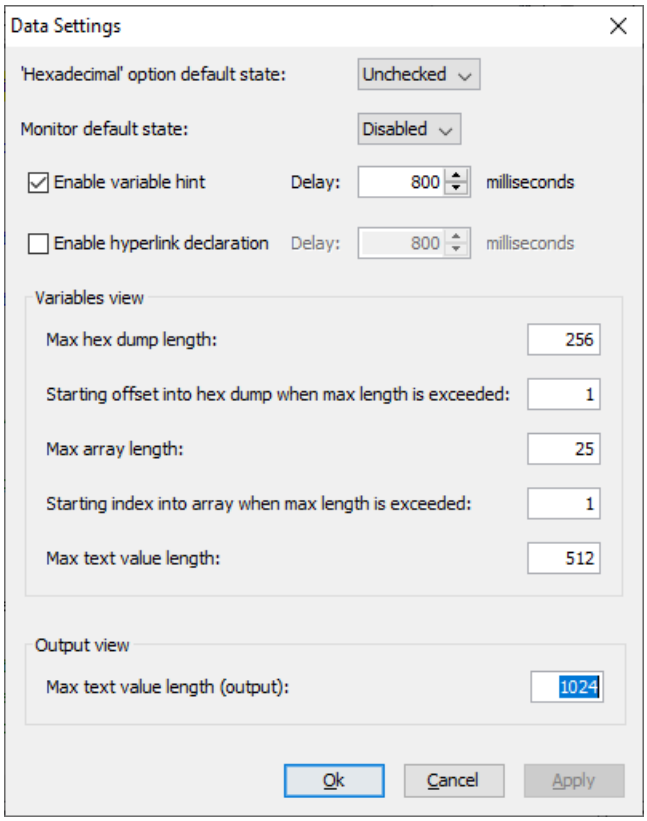
The wrk-id and wrk-name data items should be OCCURS items as they will host multiple results.

isCOBOL Debugger

The isCOBOL Debugger has been enhanced in the 2022R1 release to better manage the Output view. These are the features added:

- A new Data Setting is available to set the max text length in the Output view. This affects the double click on the data item or the command-line DISPLAY command, since both show the content of the variable in the Output view. When a variable has a long text (such as a PIC X ANY LENGTH or a PIC X(5000), it can be useful to define a limit to reduce the text shown. If the text is longer, then the characters "…" are shown after the limit. As shown in in Figure 12, *Max text value length*.

Figure 12. Max text value length.



- Data items can now be selected using the keyboard in the selected line pressing Tab or Shift+Tab. This makes it easy to use using the Debugger with the keyboard. Tab moves the selection to the next data item, while Shift+Tab moves the selection to the previous one.

For example, when selecting a source line that contains:

```
move w-customer-code    to fd-customer-code
```

it is now possible to press Shift+Tab (or Tab 2 times) to select the second data item. Pressing Ctrl+D or Ctrl+Q will cause the “Display variable” window or the “Quick Watch” window to appear with variable name fd-customer-code already entered in the field used to specify the variable name.

-Better focus management of the Debugger command line area. When the focus is moved from the Debugger command line area to another view, such as Perform stack, Breakpoints, Monitors, Current Variables, there is no need to move the focus back again to the command line area to insert a new command. In every Debugger view, when pressing an alphabetic character for commands, like D to insert the Display command or M to insert the Monitor command, the focus is automatically given to the command line area and the pressed character is input to complete the command.

isCOBOL EIS

isCOBOL EIS, Veryant's solution to write web-enabled COBOL programs, is constantly updated to provide more comprehensive web solutions. In isCOBOL 2022R1, the HTTPClient class can set and get timeouts, the $ELK directives used to generate the ServiceBridge programs have been enhanced and new configuration settings are available.

HTTPClient

HTTPClient is a class that enables COBOL programs to interact with Web Services. This class has been updated to manage the Timeout during connections and reads.

The new method signatures are shown below:

```
public void setConnectTimeout(java.lang.Double)

public void setReadTimeout(java.lang.Double)

public java.lang.Double getConnectTimeout()

public java.lang.Double getReadTimeout()
```

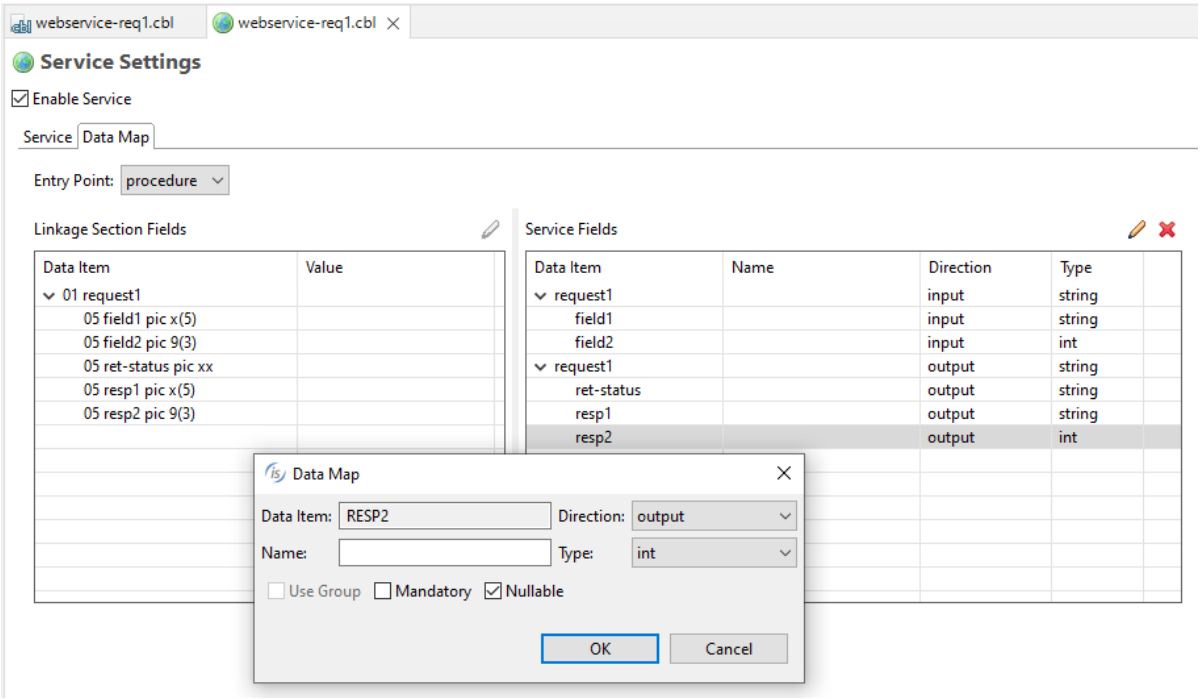
If the connect or read operation is not completed within the timeout value, an exception will be raised. This can be used to avoid having a program hangs if a URL is not responding, and without needing to add additional code to avoid this situation.

$ELK directives

When generating service bridge programs, $ELK directives help to better describe the operations and parameters that need to be passed when consuming web services. In this release, a new directive named $ELK NULLABLE is supported to set a field value to null in the Json stream. Also, the IDE Service Bridge editor has been updated to give the ability to generate this new directive in the linkage section of Cobol source.

An example is shown in Figure 13, *Nullable in Service Editor*.

Figure 13. Nullable in Service Editor.



The Nullable check-box has been checked on the output field resp2, so in the Cobol source the linkage section will be updated as:

```
$elk output
$elk nullable
     05 resp2 pic 9(3).
```

and in the generated program restwebservice-req1.cbl, the Json definition of the response will be:

```
01  response-varout identified by "Response".
  02  request1-out identified by "request1_out".
    06   RET-STATUS-outvar identified by "ret_status".
      07  ret_status-out   pic x any length.
    06   RESP1-outvar identified by "resp1" is nullable.
      07  resp1-out   pic x any length.
    06   RESP2-outvar identified by "resp2" is nullable.
      07  resp2-out   pic 9(3).
```

COBOL programs can then set null values in the field's initialization, instead of using spaces for pic x data items, or 0 in pic 9 data items.

The Json stream returned to the client consumer of this REST web service will be:

```
{
"Response":{
   "request1_out":{
             "ret_status":"OK",
             "resp1":null,
             "resp2":null
          }
       }
}
```

In addition, the existing base64Binary and hexBinary values supported in the $elk type directive (previously supported only by SOAP web services) can now also be set when using REST web services. For example, the code snippet:

```
    01 request1 identified by "request1".
   $elk input
   $elk type=base64Binary
        05 field1 pic x any length.
   $elk output
   $elk type=hexBinary
        05 field2 pic x any length.
```

defines a Json stream structure where the input to field1 contains a base64Binary parameter type, and the output to filed2 is returned as hexBinary type.

Configuration settings

New configuration settings have been implemented to customize the initialization of items in the stream structure:

iscobol.xmlstream.initialize_on_read=true to initialize the data item associated to the xml stream object

iscobol.jsonstream.initialize_on_read=true to initialize the data item associated to the json stream object

Additional improvements

isCOBOL 2022R1 improves several utilities:

- in GIFE, the Graphical Indexed and relative File Editor utility, a new configuration has been added:

  iscobol.gife.custom_font=FontName-size to customize the font used in the GIFE user interface. By default, it loads the default Font of the current LookAndFeel used, which varies based on the options –system, --nimbus, --metal, etc.

- in JDBC2FD, the utility that enables you to connect to RMDBS database and generate the COBOL copy files (.sl for the Select, .fd for the File Definition, .wrk for FD/Working structure), a new configuration has been added:

  iscobol.jdbc2fd.current_schema_only=true to load only the items of the current schema by JDBC2FD. This simplifies the filter on table names loaded after the connection.