

# Modernizing your COBOL application by using isCOBOL Compiler code injection

Author: Veryant Support

Saved From: <http://support.veryant.com/support/phpkb/question.php?ID=322>

In version 2019 R1 we introduced new compiler configuration variable to inject COBOL code in all controls of a specific type at compile time.

iscobol.compiler.gui..defaults=...

where can be any of the following: bar, bitmap, check\_box, combo\_box, date\_entry, entry\_field, frame, grid, java\_beans, label, list\_box, push\_button, radio\_button, ribbon, scroll\_bar, slider, status\_bar, tab\_control, tree\_view, web\_browser, window, tool\_bar.

This feature simplifies the modernization process for GUI applications and reduces developing efforts.

As an example, when compiling the following screen section controls:

```
01 s1.  
03 ef1 entry-field  
      line 2 col 2 size 10.  
03 ef2 entry-field  
      line 2 col 14 size 10.  
03 ef3 entry-field  
      line 2 col 26 size 10.  
03 pb1 push-button  
      line 5 col 10 size 10  
      title "Save" exception-value 1.
```

with the following compiler configuration:

```
iscobol.compiler.gui.push_button.defaults=flat, background-color -14675438  
iscobol.compiler.gui.entry_field.defaults=border-color rgb x#dae1e5, border-width (0 0 2 0 )
```

the compiler will treat the source code as if it were written as:

```
01 s1.  
03 ef1 entry-field  
      border-color rgb x#dae1e5,  
      border-width (0 0 2 0)  
      line 2 col 2 size 10.  
03 ef2 entry-field  
      border-color rgb x#dae1e5,  
      border-width (0 0 2 0)  
      line 2 col 2 size 10.  
03 ef3 entry-field  
      border-color rgb x#dae1e5,  
      border-width (0 0 2 0)  
      line 2 col 2 size 10.  
03 pb1 push-button  
      flat, background-color -14675438  
      line 5 col 10 size 10  
      title "Save" exception-value 1.
```

Code injection also affects controls created with single display statement, so that:

```
display push-button  
      line 5 col 25 size 10
```

```
title "End" exception-value 27  
handle in h-pb.
```

becomes:

```
display push-button flat, background-color -14675438  
line 5 col 25 size 10  
title "End" exception-value 27  
handle in h-pb.
```

With code injection, an entire application can be recompiled without code changes.

Changing the configuration variables can result in a completely different looking application, letting you modernize your screens without altering the source code.

**NOTE:** Code injection works by inserting the text value of the configuration variables into the source code where controls are declared or created.

That means any syntax errors in the configuration variables will result in compilation errors.

As an example, the following screen is a standard screen with slightly dated looking controls and window:

The screenshot shows a Windows-style application window titled "CUSTOMER MAINTENANCE". The window has a toolbar with "Lookup", "First", "Prev", "Next", "Last", "Save", "Delete", "Print", and "Exit".

**Customer Section:**

- Customer code: 1
- First name: Veryant LLC
- Last name: (empty)

**Address Section:**

- Street: 6390 Greenwich Drive, Ste 225
- City: San Diego
- State: CA
- Zip code: 92122

**Detail Section:**

- Gender: Male (radio button selected)
- Phone number: 619-453-0945
- Cell Phone number: (empty)

**Navigation:**

- Customer maintenance (link)

By compiling with the following compiler configuration:

```
# Compiler-regexp to remove the 3D and "ERASE" styles when displaying the  
window
```

```
iscobol.compiler-regexp=(?i)( 3-D,) " " &#92  
" (?i)( 3-D)" " " &#92  
" (?i)( ERASE,)" " " &#92  
" (?i)( ERASE)" " "
```

```

##### code injection for controls #####
# add the gradient color on all windows
iscobol.compiler.gui.window.defaults= &#92
gradient-color-1 rgb x#FFFFFF &#92
gradient-color-2 rgb x#F2F5F9  &#92
gradient-orientation gradient-northeast-to-southwest

# add the transparent style to all labels, check-boxes and radio-buttons
iscobol.compiler.gui.label.defaults= transparent
iscobol.compiler.gui.check_box.defaults= transparent
iscobol.compiler.gui.radio_button.defaults= transparent
iscobol.compiler.gui.frame.defaults= transparent

# set the white color for all toolbars
iscobol.compiler.gui.tool_bar.defaults= background-Color rgb x#FFFFFF &#92
                                         foreground-Color rgb x#000000

# add the flat style to all push buttons
iscobol.compiler.gui.push_button.defaults=flat

# add the underline style to all entry-fields
iscobol.compiler.gui.entry_field.defaults = border-width (0, 0, 2, 0) &#92
                                         border-color rgb x#DAE1E5

```

The screen is transformed to this more modern looking screen.

The screenshot shows a customer maintenance application window titled "CUSTOMER MAINTENANCE". The window has a standard OS X-style title bar with minimize, maximize, and close buttons. Below the title bar is a toolbar with icons for "Lookup", "First", "Prev", "Next", "Last", "Save", "Delete", "Print", and "Exit".

The main content area is divided into sections:

- Customer:** Contains fields for "Customer code" (value 1), "First name" (value "Veryant LLC"), and "Last name" (empty).
- Address:** Contains fields for "Street" (value "6390 Greenwich Drive, Ste 225"), "City" (value "San Diego"), "State" (value "CA"), and "Zip code" (value "92122").
- Detail:** Contains gender selection ("Gender: Male (radio selected) Female (radio unselected)"), phone number ("Phone number: 619-453-0945"), and cell phone number ("Cell Phone number: [empty]").

At the bottom left is a navigation bar with a back arrow and the text "Customer maintenance".