COBGDB - The GnuCOBOL TUI DEBUGGER / ANIMATOR

HOW TO USE COBGDB

Table of Contents

1. Introduction

COBGDB is a TUI (Text User Interface) application, programmed in C, designed to assist in animate and debugging GnuCOBOL programs using **GDB** (the GNU Debugger at https://www.gnu.org/software/gdb/).

The COBGDB project is hosted at ***https://github.com/marcsosduma/cobgdb***

Very important: you don't need to know how to use the GDB product and its many commands (https://www.sourceware.org/gdb/).
COBGDB has its own user interface (described in this document) that is very simple to use and is responsible for interfacing the underlying GDB which is the real debug and animate engine but operates practically in a transparent way to the GnuCOBOL developer.

The COBGDB application is based on the extension for Visual Studio Code (VSCode) created by Oleg Kunitsyn, which can be found on GitHub: *https://github.com/OlegKunitsyn/gnucobol-debug.*

At https://github.com/marcsosduma/cobgdb in the Windows subdirectory, the executable program cobgdb.exe for this operating system is available and ready to use.

To compile COBGDB from C source code on Windows, you can use MinGW.
The Makefile is configured to generate the program **`cobgdb.exe`** for both Windows and Linux.

1.1. Installing the debugger COBGDB on Windows

On Windows, just download **`cobgdb.exe`** from following folder:
https://github.com/marcsosduma/cobgdb/tree/main/windows .

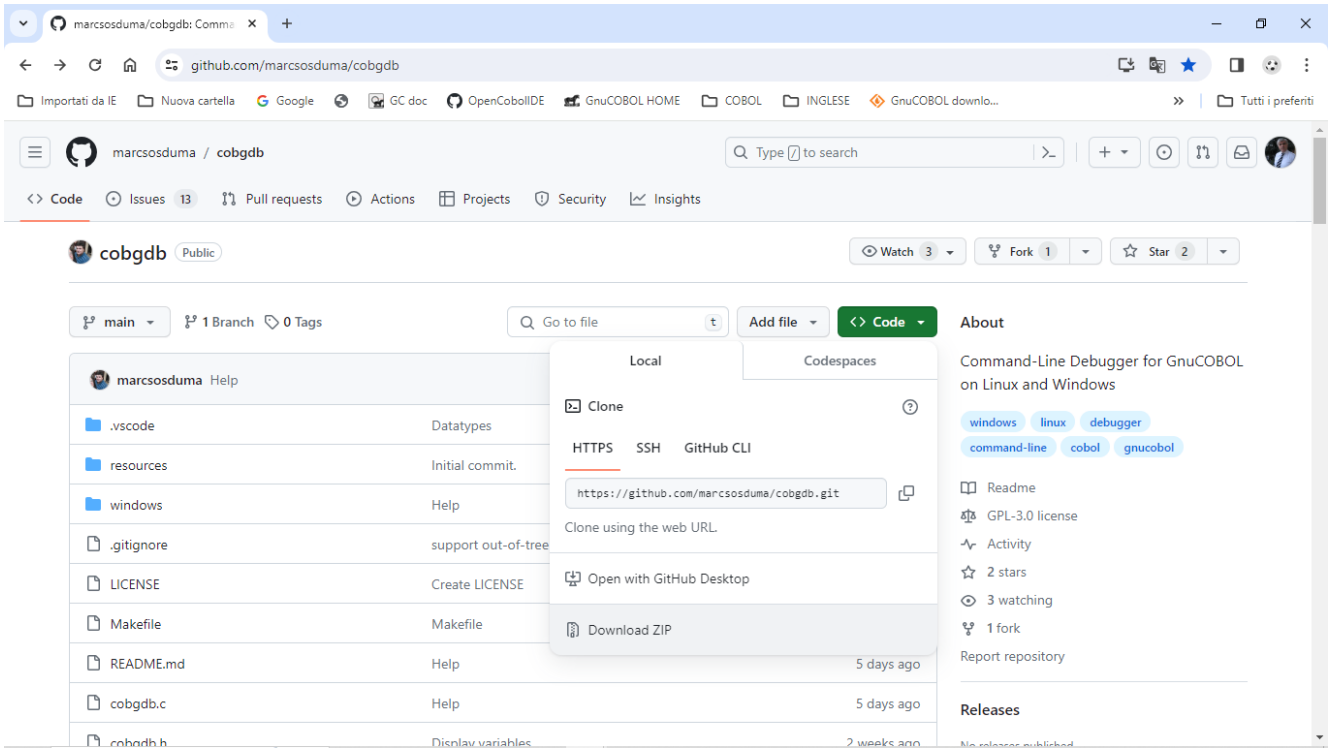
As an example you can put **`cobgdb.exe`** into the "**`bin`**" folder of your GnuCOBOL installation (the same folder where the GnuCOBOL compiler **`cobc.exe`** is located)

or

first install MinGW (Minimalist GNU for Windows).
Then execute the make ('mingw32-make' for Windows) command to compile the code from C source.

Note: if you have security problems downloading an .exe file then you can try download the entire repository with the following github button **<> Code v** --> *Download ZIP*

Then unzip the file and copy cobgdb.exe to the "bin" folder of your GnuCOBOL installation (the same folder where the GnuCOBOL compiler cobc.exe is located)



1.2. Compile and Debug GnuCOBOL programs

Compile and run a debugging session of the sample program using the following command:

```
cobgdb customer.cob -x -lpdcurses
```

Source code of customer.cob used also for following tutorial is at:
https://github.com/marcsosduma/cobgdb/tree/main/windows

Note: '**`-lpdcurses`**' is an instance of an argument that can be indirectly passed to '**`cobc`**' by '**`cobgdb`**,' even if it is not used by '**`cobgdb`**' itself.

or, other example for **`cobc`** parameters , use : **`cobgdb customer.cob -x -Tcustomer.txt`** .(-T creates a compilation list output into **`customer.txt`** file)

COBGDB takes one or more programs with COB or CBL extension as parameters and runs the GnuCOBOL compiler with the following format:

```
cobc -g -fsource-location -ftraceall -v -free -O0 -x prog.cob prog2.cob ...
```

To debug multiple programs, use COBGDB with the following syntax:

```
cobgdb prog.cob subprog1.cob subprog2.cob . . .
```

This will create a single prog.exe executable.

You can run GDB/GDBSERVER remotely using the "A" (Attach) key.
COBGDB will prompt you to provide the server and port in the format server: port or the PID of the application.
Example:

- localhost: 5555
- 9112

1.3. Main Commands

Cmd		Description	
?	Help	Show the HELP window	
R	Run	This is the first command to always use to start a debugging session. Start and Run the program until a breakpoint is encountered, also when, by default, the first Breakpoint is at first program statement.	
B	Breakpoint	Set or Unset a Breakpoint at a specific line of the Procedure Division code.	
C	Cursor	Runs the program until it reaches the selected line at Cursor location.	
N	Next	Runs the program until the next line but does not enter a subroutine executed by CALL or PERFORM.	
S	Step	Runs the program until the next line. If needed it goes into a subroutine executed by CALL or PERFORM.	
G	Go	Continues the program execution until it encounters a stopping point: a breakpoint, the end of the program, or the return from a subroutine (PERFORM / CALL).	
J	Jump	Ask for a line number and Runs the program until it reaches that line.	
V	Variables	Displays a window with a list of all variables and their content for the running program. From this window you can also change the content of variables.	
H	sHow	Displays a window with a list of variables and its content from the cursor selected line. From this window you can also edit / change the content of variables from the selected line. Right-click on a row is same as command "H".	
D	Display	Automatic Display of variables of current and previous statement in execution during a debugging / animation session is settled to OFF or ON. At program start is OFF.	
F	File	When COBGDB is executed with more than one program, allows selecting one of those source file to manage debugging commands.	
A	Attach	Attach to GDBSERVER or to an Application PID.	
W	Window	Switch between two window size: 24 rows x 80 cols or 34 rows x 132 cols.	
Q	Quit	Quits (ends) the debugging / animation session and the program (or programs).	

2. Tutorial - Sample Debugging Session

Following tutorial is on a Windows 10 platform using follwing version of GnuCOBOL:

```
cobc (GnuCOBOL) 3.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Keisuke Nishida, Roger While, Ron Norman, Simon Sobisch, Edward Hart
Built Jul 28 2023 16:07:38
Packaged Jul 28 2023 16:58:47 UTC
C version (MinGW) "13.1.0"
```

Downloaded from *https://www.arnoldtrembley.com/GnuCOBOL.htm*

====== Version 3.2 ======
GnuCOBOL 3.2 (28Jul2023) MSYS2 64-bit GC32M-BDB-x64.7z -- MSYS2 64-bit GnuCOBOL 3.2 Final release **with full debugging support**. (95.4 Megabytes).
GnuCOBOL 3.2 (28Jul2023) MSYS2 32-bit GC32M-BDB-x32.7z -- MSYS2 32-bit GnuCOBOL 3.2 Final release **with full debugging support**. (96.1 Megabytes).

After executing **cobgdb customer.cob -x -lpdcurses** the application automatically insert a Breakpoint at first executable program statement of PROCEDURE DIVISON (see the B symbol at left of line 103 in this sample) and displays following screen:

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter                    ▶ → ↓ ↑ ■ D ?
  100                 10 COLUMN PLUS 2 TO WS-ERROR.
  101
  102           PROCEDURE DIVISION.
B 103           001-START.
  104               SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
  105               SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
  106               SET ENVIRONMENT 'ESCDELAY' TO '25'
  107               *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
  108               *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
  109               ACCEPT WS_NUMR FROM LINES
  110               ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
  111               PERFORM 007-OPEN-FILES
  112               PERFORM UNTIL E-EXIT
  113                   MOVE "MENU" TO WS-OP
  114                   MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
  115                   MOVE SPACES TO WS-CHOICE
  116                   DISPLAY SS-CLS
  117                   ACCEPT SS-MENU
  118
  119                   EVALUATE TRUE
  120                       WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
run
```

You can scroll the source code window with cursor keys UP and DOWN, PG UP and PG DOWN or with mouse wheel or with mouse left click on the right scroll bar. Use cursor RIGHT and cursor LEFT to scroll horizontally,

In the upper right window corner there is a "button bar" where you can find some buttons (symbols):

> → ↓ ↑ ■ D ?

>	=	**Run** command
→	=	**Next** command
↓	=	**Step** command
↑	=	**Go** command
■	=	**Quit** command
D	=	**Display** command
?	=	**Help** command

when you hover over one of these symbol, you get the corresponding command description (like a tooltip) displayed at the bottom left of the screen.

2.1. Help Command

Type **?** (key) HELP command or left click with mouse on the **?** button:

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                 10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102          PROCEDURE DIVISION.
B 103          001-START.
 104              SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105              SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106              SET ENVIRONMENT 'ESCDELAY' TO '25'
 107              *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 108              *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
 109              ACCEPT WS_NUMR FROM LINES
 110              ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
 111              PERFORM 007-OPEN-FILES
 112              PERFORM UNTIL E-EXIT
 113                  MOVE "MENU" TO WS-OP
 114                  MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
 115                  MOVE SPACES TO WS-CHOICE
 116                  DISPLAY SS-CLS
 117                  ACCEPT SS-MENU
 118
 119                  EVALUATE TRUE
 120                      WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
help
```

the HELP window is displayed

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                 10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102          PROCEDURE DIVISION.
B 103          001-START.
 104              SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 10 COBGDB HELP
 10                    COBGDB - Commands
 10
 10 B - Breakpoint: toggles the breakpoint at the current selected line    ATU
 10     (can also be done with the mouse).
 11 R - Run: runs the program from the first statement until a breakpoint
 11     is encountered.
 11 C - Cursor or Continue: runs the program until it reaches the
 11     selected line.
 11 J - Jump: runs the program until it reaches the specified line.
 11 N - Next: runs the program until the next line but does not enter
 11
 117                  ACCEPT SS-MENU
 118
 119                  EVALUATE TRUE
 120                      WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
help
```

scroll the Help window with cursor keys UP and DOWN or mouse wheel.
Use ESC or Enter or left click to exit from this HELP window and return to debugging session.

2.2. Run Command

To start executing the program and the debugging session from first program statement you always must use the "**R**" command (key) or left click with mouse on the **>** **Run** button

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
  100                 10 COLUMN PLUS 2 TO WS-ERROR.
  101
  102         PROCEDURE DIVISION.
B 103         001-START.
  104             SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
  105             SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
  106             SET ENVIRONMENT 'ESCDELAY' TO '25'
  107             *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
  108             *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
  109             ACCEPT WS_NUMR FROM LINES
  110             ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
  111             PERFORM 007-OPEN-FILES
  112             PERFORM UNTIL E-EXIT
  113                 MOVE "MENU" TO WS-OP
  114                 MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
  115                 MOVE SPACES TO WS-CHOICE
  116                 DISPLAY SS-CLS
  117                 ACCEPT SS-MENU
  118
  119                 EVALUATE TRUE
  120                     WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
run
```

cobgdb opens the program terminal window (the application will run in this separate window.).

```
C:\GC-AWORK\customer.exe
```

go back to the COBGDB screen, and you see a > green symbol on the left of statement where initial B Breakpoint is present (in our example is at line 103 B>103):

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102          PROCEDURE DIVISION.
B>103         001-START.
 104              SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105              SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106              SET ENVIRONMENT 'ESCDELAY' TO '25'
 107              *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 108              *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
 109              ACCEPT WS_NUMR FROM LINES
 110              ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
 111              PERFORM 007-OPEN-FILES
 112              PERFORM UNTIL E-EXIT
 113                  MOVE "MENU" TO WS-OP
 114                  MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
 115                  MOVE SPACES TO WS-CHOICE
 116                  DISPLAY SS-CLS
 117                  ACCEPT SS-MENU
 118
 119                  EVALUATE TRUE
 120                      WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Debugging
```

From that moment on, you can use all the commands (keys) or corresponding buttons to "animate" and debug the application, example: "**S**" (Step), "**N**" (Next), "**G**" (Go) and so on.

During the source code animation, the debugger can automatically shows some pop-up windows with variables content from the line in execution.

In order not to slow down the debugging / animation of the program unnecessarily, the automatic display of all variables of the statement in execution is disabled by default at program startup.
To activate the Automatic Display of all variables of the statement in execution use the "D" command.

2.3. Step Command

Proceed with **S** (Step) command or left click with mouse the ↓ button:.

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102        PROCEDURE DIVISION.
B>103       001-START.
 104            SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105            SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106            SET ENVIRONMENT 'ESCDELAY' TO '25'
 107            *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 108            *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
 109            ACCEPT WS_NUMR FROM LINES
 110            ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
 111            PERFORM 007-OPEN-FILES
 112            PERFORM UNTIL E-EXIT
 113                MOVE "MENU" TO WS-OP
 114                MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
 115                MOVE SPACES TO WS-CHOICE
 116                DISPLAY SS-CLS
 117                ACCEPT SS-MENU
 118
 119                EVALUATE TRUE
 120                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
step
```

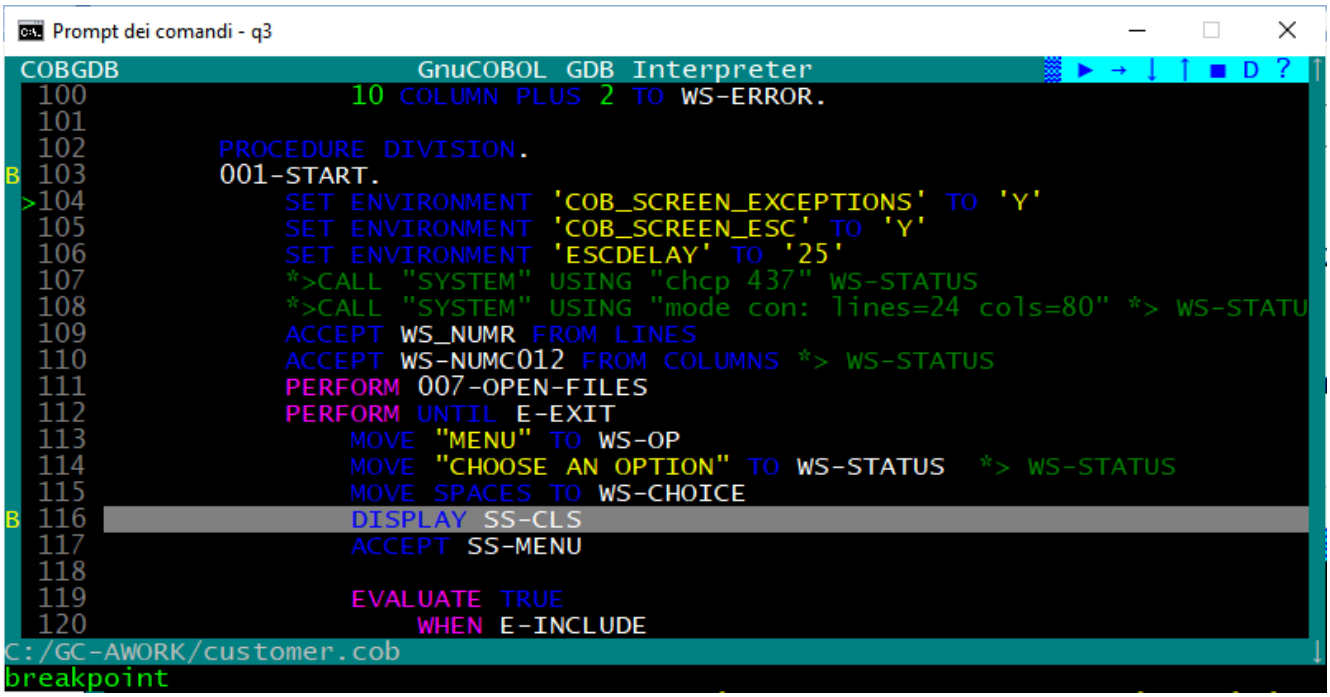
the **>104** green symbol now is on the following line 104:

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102        PROCEDURE DIVISION.
B 103       001-START.
 >104           SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105            SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106            SET ENVIRONMENT 'ESCDELAY' TO '25'
 107            *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 108            *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
 109            ACCEPT WS_NUMR FROM LINES
 110            ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
 111            PERFORM 007-OPEN-FILES
 112            PERFORM UNTIL E-EXIT
 113                MOVE "MENU" TO WS-OP
 114                MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
 115                MOVE SPACES TO WS-CHOICE
 116                DISPLAY SS-CLS
 117                ACCEPT SS-MENU
 118
 119                EVALUATE TRUE
 120                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Debugging
```

Now you can proceed with S command or N command or as an example:

- Scroll with cursor down to select line 116 of Procedure Division and type "B" (to set a Breakpoint), (you also can simply click with mouse left button on the 116 row number)

- The application displays a "**B**" on the left of the line (type B again - or re-click - when you want to delete the Breakpoint, not do that at this moment)



2.4. Go Command

Type **G** (Go) or left click with mouse the ↑ button to execute the program until a B Breakpoint is detected: the system reach the second breakpoint at line 116 and displays a green > symbol to the left of the line to be executed, see following screen:

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102         PROCEDURE DIVISION.
B 103         001-START.
 104             SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105             SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106             SET ENVIRONMENT 'ESCDELAY' TO '25'
 107             *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 108             *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
 109             ACCEPT WS_NUMR FROM LINES
 110             ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
 111             PERFORM 007-OPEN-FILES
 112             PERFORM UNTIL E-EXIT                    WS-STATUS
 113                 MOVE "MENU" TO WS-OP                "CHOOSE AN OPTION"
 114                 MOVE "CHOOSE AN OPTION" TO WS-STATUS
 115                 MOVE SPACES TO WS-CHOICE
B>116                DISPLAY SS-CLS
 117                 ACCEPT SS-MENU
 118                                                     WS-MODULE
 119                 EVALUATE TRUE                       "CUSTOMERS - MENU"
 120                     WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Debugging
```

2.5. Display Variables

The debugger / animator can do an automatic display of all variables content from the statement executed and from the previous statement.
This can slow down the animation / execution and it may not be useful to have it activated everywhere.
Type **D** (Display Variables) command to enable or disable this display at your convenience.
Alternatively, click on the "D" symbol in the top right bar of the screen.
By moving the mouse over the "D" symbol in the bar you will see a message with the current status of this option displayed at the bottom left of screen: Display of Variables: ON or Display of Variables: OFF.
At the beginning of the program this option is always set to OFF to speed up the animation.
When you have reached the point of the program that interests you, you can click on "D", or type the command "D", to activate the automatic display of all the variables of the statement in execution (black windows) and of the previous statement (blue windows).

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter                    ▶ → ↓ ↑ ■ D ?
  100                 10 COLUMN PLUS 2 TO WS-ERROR.
  101
  102           PROCEDURE DIVISION.
B 103           001-START.
  104               SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
  105               SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
  106               SET ENVIRONMENT 'ESCDELAY' TO '25'
  107               *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
  108               *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
  109               ACCEPT WS_NUMR FROM LINES
  110               ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
  111               PERFORM 007-OPEN-FILES
  112               PERFORM UNTIL E-EXIT
  113                   MOVE "MENU" TO WS-OP
  114                   MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
  115                   MOVE SPACES TO WS-CHOICE
  116                   DISPLAY SS-CLS
  117                   ACCEPT SS-MENU
  118
  119                   EVALUATE TRUE
  120                       WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
display of variables: OFF
```

2.6. Show Command

typing the 'H' command (key) allows you to view the variables on the cursor selected (highlighted) line.

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102        PROCEDURE DIVISION.
B 103        001-START.
 104            SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105            SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106            SET ENVIRONMENT 'ESCDELAY' TO '25'
 107      Show Line Variables
 108      -SS-CLS: "                                                  S-STATU
 109       -SS-CLS: "
 110       -Implicit FILLER: ""
 111       -Implicit FILLER: ""
 112       -Implicit FILLER: ""
 113       -Implicit FILLER: "                                  "     ION"
 114       -SS-STATUS: "                                  "
 115       -Implicit FILLER: "                                  "
B>116
 117                ACCEPT SS-MENU
 118                                                        WS-MODULE
 119                EVALUATE TRUE                           "CUSTOMERS - MENU"
 120                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
E - Edit variable
```

Display the content of variables also clicking right mouse button on a source line, example click right mouse button on line 114 will execute the H command on that line and give you:

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102        PROCEDURE DIVISION.
B 103        001-START.
 104            SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105            SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106            SET ENVIRONMENT 'ESCDELAY' TO '25'
 107      Show Line Variables
 108      -WS-STATUS: "CHOOSE AN OPTION                    "          S-STATU
 109       -WS-STATUS: "CHOOSE AN OPTION                    "
 110                                                                  ION"
 111            PERFORM 007-OPEN-FILES
 112            PERFORM UNTIL E-EXIT
 113                MOVE "MENU" TO WS-OP
 114                MOVE "CHOOSE AN OPTION" TO WS-STATUS   *> WS-STATUS
 115                MOVE SPACES TO WS-CHOICE
B>116                DISPLAY SS-CLS
 117                ACCEPT SS-MENU
 118                                                        WS-MODULE
 119                EVALUATE TRUE                           "CUSTOMERS - MENU"
 120                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
E - Edit variable
```

Now at lower left line on screen is a message: E = Edit (change) the variable value.

2.6.1. Edit subCommand

Scroll with cursor key UP and DOWN (or with mouse wheel). to select one of the variable.
The key **E** can be used to edit the content of the highlighted variable.
Change the value and type Enter to confirm canges or use ESC to exit without changes :

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102         PROCEDURE DIVISION.
B 103         001-START.
 104             SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105             SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106             SET ENVIRONMENT 'ESCDELAY' TO '25'
 107      Edit Variable
 108      WS-STATUS: xxxxSE AN OPTION                                  S-STATU
 109
 110                                                                   ION"
 111             PERFORM 007-OPEN-FILES
 112             PERFORM UNTIL E-EXIT
 113                 MOVE "MENU" TO WS-OP
 114                 MOVE "CHOOSE AN OPTION" TO WS-STATUS   *> WS-STATUS
 115                 MOVE SPACES TO WS-CHOICE
B>116                DISPLAY SS-CLS
 117                 ACCEPT SS-MENU
 118                                                          WS-MODULE
 119                 EVALUATE TRUE                            "CUSTOMERS - MENU"
 120                     WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
```

Resulting:

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102         PROCEDURE DIVISION.
B 103         001-START.
 104             SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105             SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106             SET ENVIRONMENT 'ESCDELAY' TO '25'
 107      Show Line Variables
 108      -WS-STATUS: "xxxxSE AN OPTION                 "              S-STATU
 109       -WS-STATUS: "xxxxSE AN OPTION                 "
 110                                                                   ION"
 111             PERFORM 007-OPEN-FILES
 112             PERFORM UNTIL E-EXIT
 113                 MOVE "MENU" TO WS-OP
 114                 MOVE "CHOOSE AN OPTION" TO WS-STATUS   *> WS-STATUS
 115                 MOVE SPACES TO WS-CHOICE
B>116                DISPLAY SS-CLS
 117                 ACCEPT SS-MENU
 118                                                          WS-MODULE
 119                 EVALUATE TRUE                            "CUSTOMERS - MENU"
 120                     WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
E - Edit variable
```

2.7. Variable Command

Type the **V** command (key) to display the list of all program variables:

```
Prompt dei comandi - q3
COBGDB - (R)return (ENTER)expand/contract (E)edit var
 RETURN-CODE: 0
+FILE1 Record: "\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
+WS-MODULE: "CUSTOMERS - MENU                    "
+WS-CHOICE: " "
+FS-STAT: "00"
+WS-ERROR: " "
+WS_NUMR: "025"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "xxxxSE AN OPTION                    "
+WS-ERRMSG: "
+SS-CLS: "                                                                    "
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEX - EXITCHOICE:  "
+SS-RECORD-SCREEN: "  PHONE:              NAME:
+SS-ERROR: "
+COB-CRT-STATUS: "0000"

Debugging
```

2.7.1. Enter subCommand

Scroll with cursor key UP and DOWN (or with mouse wheel) in this list to the variable WS-MODULE and type Enter: the application opens and displays its subfields WS-MODULE and WS-OP

```
Prompt dei comandi - q3
COBGDB - (R)return (ENTER)expand/contract (E)edit var
 RETURN-CODE: 0
+FILE1 Record: "\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
-WS-MODULE: "CUSTOMERS - MENU                    "
  WS-MODULE: "CUSTOMERS - MENU                    "
  WS-OP: "MENU                    "
+WS-CHOICE: " "
+FS-STAT: "00"
+WS-ERROR: " "
+WS_NUMR: "025"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "xxxxSE AN OPTION                    "
+WS-ERRMSG: "
+SS-CLS: "                                                                    "
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEX - EXITCHOICE:  "
+SS-RECORD-SCREEN: "  PHONE:              NAME:
+SS-ERROR: "
+COB-CRT-STATUS: "0000"

Debugging
```

select WS-MODULE subfield

```
Prompt dei comandi - q3
COBGDB - (R)return (ENTER)expand/contract (E)edit var
 RETURN-CODE: 0
+FILE1 Record: "\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
-WS-MODULE: "CUSTOMERS - MENU                    "
  WS-MODULE: "CUSTOMERS - MENU                    "
  WS-OP: "MENU                    "
+WS-CHOICE: " "
+FS-STAT: "00"
+WS-ERROR: " "
+WS_NUMR: "025"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "xxxxSE AN OPTION                    "
+WS-ERRMSG: "                                                                    "
+SS-CLS: "                                                                        "
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEX - EXITCHOICE:  "
+SS-RECORD-SCREEN: "  PHONE:              NAME:
+SS-ERROR: "
+COB-CRT-STATUS: "0000"

Debugging
```

2.7.2. Edit subCommand

Now you can select WS-MODULE subfield whit cursor DOWN or mouse wheel and type "**E**" (Edit)
COBGDB shows a Edit Variable window:

```
Prompt dei comandi - q3
COBGDB - (R)return (ENTER)expand/contract (E)edit var
 RETURN-CODE: 0
+FILE1 Record: "\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
-WS-MODULE: "CUSTOMERS - MENU                    "
  WS-MOD Edit Variable
  WS-OP: WS-MODULE: CUSTOMERS - MENU
+WS-CHOI
+FS-STAT: "00"
+WS-ERROR: " "
+WS_NUMR: "025"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "xxxxSE AN OPTION                    "
+WS-ERRMSG: "                                                                  "
+SS-CLS: "                                                                     "
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEX - EXITCHOICE:  "
+SS-RECORD-SCREEN: "  PHONE:              NAME:
+SS-ERROR: "
+COB-CRT-STATUS: "0000"

Debugging
```

Change "CUSTOMERS" to "TEST" and type Enter to change the value (use ESC to exit without changes) :

```
Prompt dei comandi - q3
COBGDB - (R)return (ENTER)expand/contract (E)edit var
 RETURN-CODE: 0
+FILE1 Record: "\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
-WS-MODULE: "CUSTOMERS - MENU                    "
  WS-MOD Edit Variable
  WS-OP: WS-MODULE: TEST     - MENU
+WS-CHOI
+FS-STAT: "00"
+WS-ERROR: " "
+WS_NUMR: "025"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "xxxxSE AN OPTION                    "
+WS-ERRMSG: "                                                                  "
+SS-CLS: "                                                                     "
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEX - EXITCHOICE:  "
+SS-RECORD-SCREEN: "  PHONE:              NAME:
+SS-ERROR: "
+COB-CRT-STATUS: "0000"

Debugging
```

2.7.3. Return subCommand

WS-Module has new value.

Now type "**R**" (Return) to go back to the debugging session:

```
Prompt dei comandi - q3
COBGDB - (R)return (ENTER)expand/contract (E)edit var
 RETURN-CODE: 0
+FILE1 Record: "\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
-WS-MODULE: "TEST       - MENU                    "
  WS-MODULE: "TEST       - MENU                    "
  WS-OP: "MENU                    "
+WS-CHOICE: " "
+FS-STAT: "00"
+WS-ERROR: " "
+WS_NUMR: "025"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "xxxxSE AN OPTION                "
+WS-ERRMSG: "                                                                "
+SS-CLS: "                                                                   "
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEX - EXITCHOICE:  "
+SS-RECORD-SCREEN: "  PHONE:              NAME:
+SS-ERROR: "
+COB-CRT-STATUS: "0000"

Debugging
```

2.8. Step Command

now you are back in the main debugging window:

Type **S** (Step) command or left click with mouse the ↓ button to execute the DISPLAY statement at line 116

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102        PROCEDURE DIVISION.
B 103        001-START.
 104            SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105            SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106            SET ENVIRONMENT 'ESCDELAY' TO '25'
 107            *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 108            *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
 109            ACCEPT WS_NUMR FROM LINES                 WS-STATUS
 110            ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STA  "xxxxSE AN OPTION"
 111            PERFORM 007-OPEN-FILES
 112            PERFORM UNTIL E-EXIT
 113                MOVE "MENU" TO WS-OP
 114                MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
 115                MOVE SPACES TO WS-CHOICE
B>116               DISPLAY SS-CLS
 117                ACCEPT SS-MENU
 118                                                      WS-MODULE
 119                EVALUATE TRUE                         "TEST       - MENU"
 120                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
step
```

in the other application window you can see the result of DISPLAY statement

```
C:\GC-AWORK\customer.exe
TEST       - MENU

xxxxSE AN OPTION
```

go back to debugging window and now the ACCEPT statement will be executed with S command

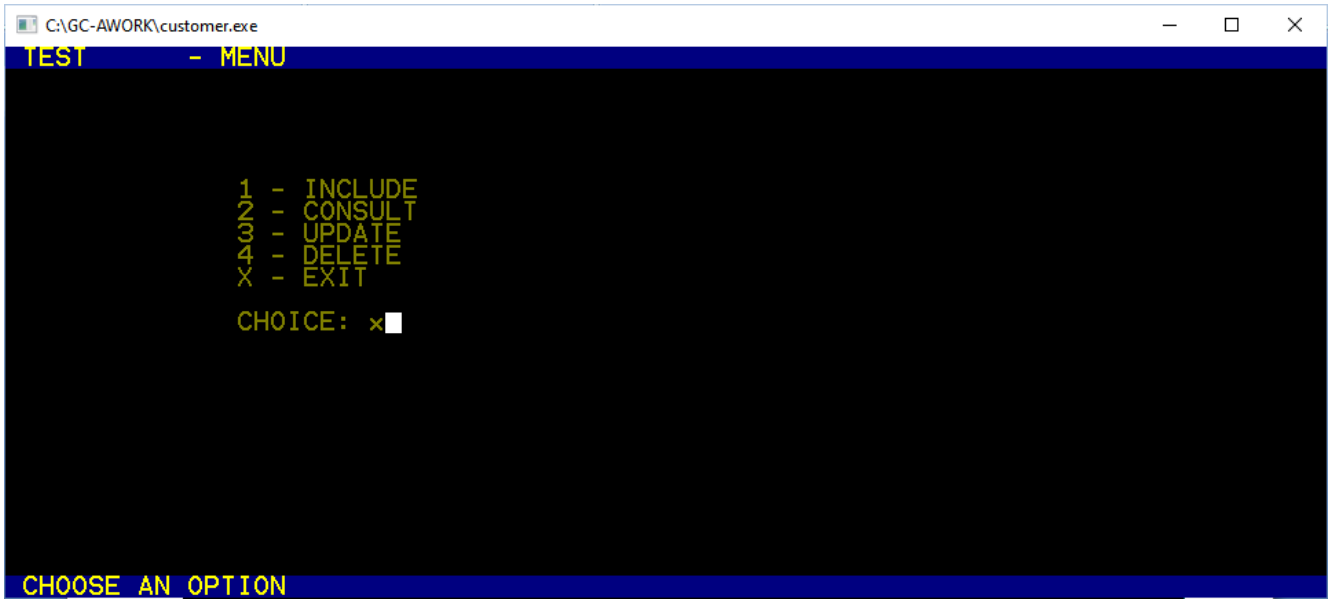
Type **S** (Step) command or leftclick with mouse the ↓ button again to execute ACCEPT statement at line 117:

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102        PROCEDURE DIVISION.
B 103        001-START.
 104            SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS'
 105            SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106            SET ENVIRONMENT 'ESCDELAY' TO '25'
 107            *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 108            *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
 109            ACCEPT WS_NUMR FROM LINES
 110            ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STA
 111            PERFORM 007-OPEN-FILES
 112            PERFORM UNTIL E-EXIT
 113                MOVE "MENU" TO WS-OP
 114                MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
 115                MOVE SPACES TO WS-CHOICE
B 116                DISPLAY SS-CLS
 >117                ACCEPT SS-MENU
 118
 119                EVALUATE TRUE
 120                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
step
WS-CHOICE
""
WS-MODULE
"TEST      - MENU"
WS-STATUS
"CHOOSE AN OPTION"
```

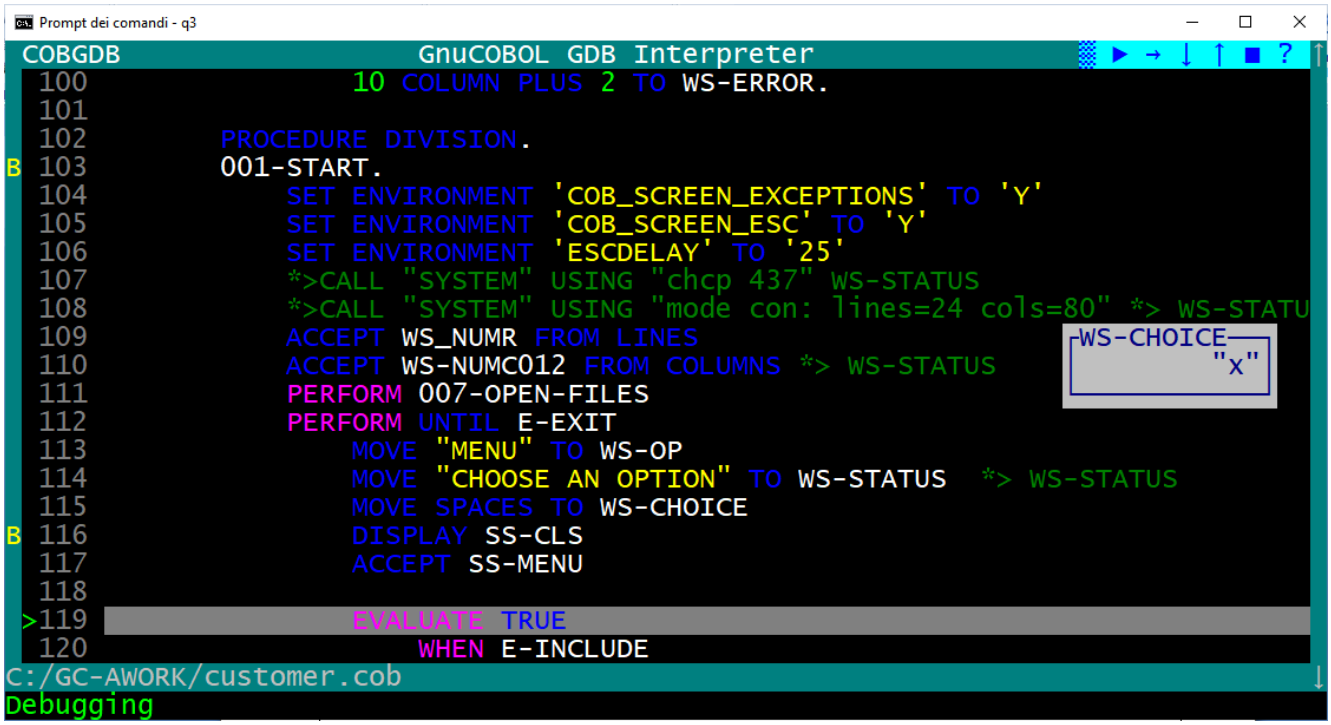
A red **!** quotation mark appears on the line **!117** .
This means that application is running and a user action is required at application window.

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102        PROCEDURE DIVISION.
B 103        001-START.
 104            SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105            SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106            SET ENVIRONMENT 'ESCDELAY' TO '25'
 107            *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 108            *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
 109            ACCEPT WS_NUMR FROM LINES
 110            ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
 111            PERFORM 007-OPEN-FILES
 112            PERFORM UNTIL E-EXIT
 113                MOVE "MENU" TO WS-OP
 114                MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
 115                MOVE SPACES TO WS-CHOICE
B 116                DISPLAY SS-CLS
 !117                ACCEPT SS-MENU
 118
 119                EVALUATE TRUE
 120                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Running
```

The application is running in a separate window.
The 'Accept' command switches the focus to application windows.
After a user action on application screen (ex type the "X" choice and Enter) it is necessary to click again on the 'debugger' window to continue debugging.



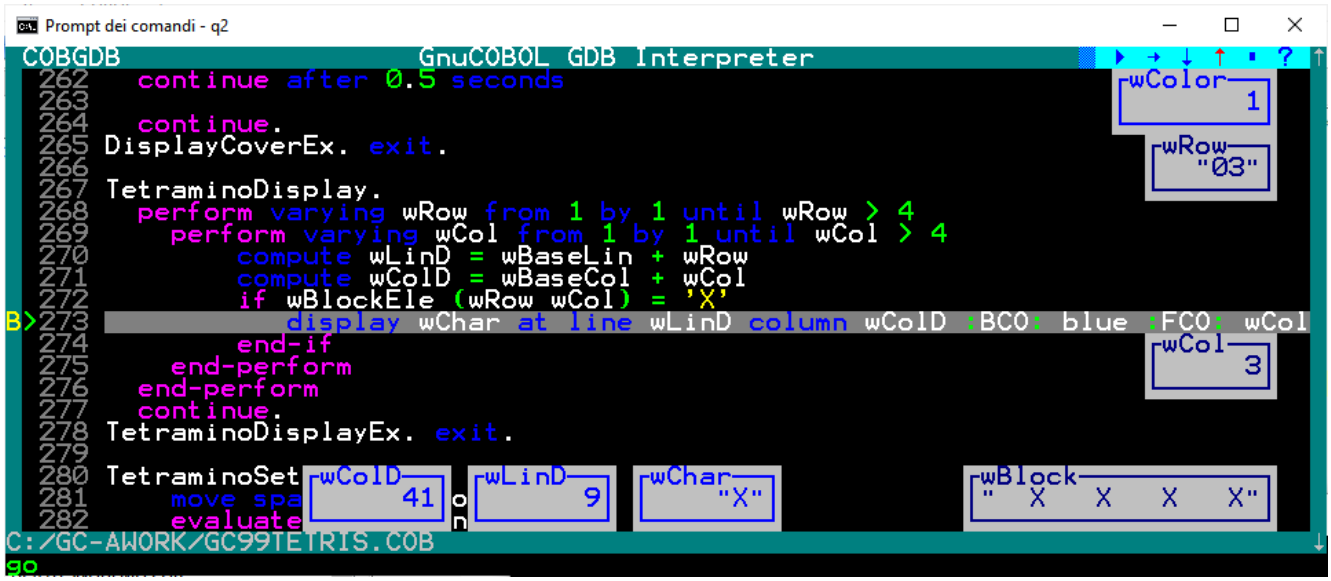
go back to debugger window: the ACCEPT statement has been executed:



2.9. Pop-up Variable windows

During a debugging session COBGDB shows variable content.
Blue frame and values : variables of executing cobol statement
Black frame and values : variables of last executed cobol statement.

Sample:



2.10. File Command

To show this command we use following sample:

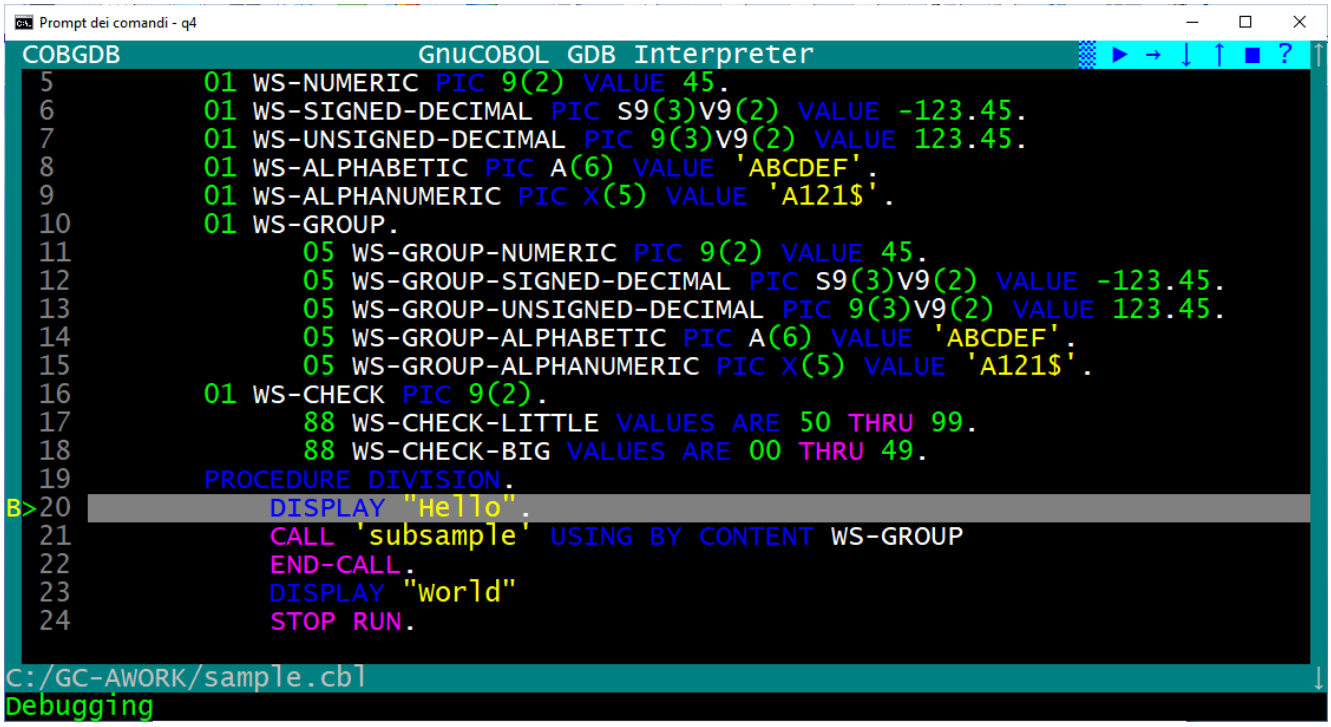
```
cobgdb sample.cbl subsample.cbl subsubsample.cbl -x -lpdcurses
```

where sample.cbl is the main program; it calls
--> subsample.cbl; it calls
--> subsubsample.cbl

Source code is at https://github.com/marcsosduma/cobgdb/tree/main/resources.
This will create a single sample.exe executable.

This example shows that when you need to debug only subsample.cbl or only subsubsample.cbl you need to execute COBGDB with all three programs.

COBGDB sets the B breakpoint at first executable statement of first program "sample.cbl".
here use the R Run command to start the debugging session.



Now you can type the **F File** command and you will have the "Source Files" window.
In this sample we select the second program in the list (subsample.cbl) and type Enter.

```
Prompt dei comandi - q4
COBGDB                    GnuCOBOL GDB Interpreter
 5          01 WS-NUMERIC PIC 9(2) VALUE 45.
 6          01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
 7          01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
 8          01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
 9          01 WS-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
10      -Source Files-
11      C:/GC-AWORK/sample.cbl
12      C:/GC-AWORK/subsample.cbl                                  .45.
13      C:/GC-AWORK/subsubsample.cbl                               .45.
14
15
16
17
18
19
B>20
21
22              END-CALL.
23              DISPLAY "World"
24              STOP RUN.
C:/GC-AWORK/sample.cbl
Debugging
```

COBGDB shows the selected program source code where in this sample we type a B command at line 14.

```
Prompt dei comandi - q4
COBGDB                    GnuCOBOL GDB Interpreter
 1          IDENTIFICATION DIVISION.
 2          PROGRAM-ID. subsample.
 3          ENVIRONMENT DIVISION.
 4          DATA DIVISION.
 5          WORKING-STORAGE SECTION.
 6          LINKAGE SECTION.
 7          01 WS-GROUP.
 8                05 WS-GROUP-NUMERIC PIC 9(2).
 9                05 WS-GROUP-SIGNED-DECIMAL PIC S9(3)V9(2).
10                05 WS-GROUP-UNSIGNED-DECIMAL PIC 9(3)V9(2).
11                05 WS-GROUP-ALPHABETIC PIC A(6).
12                05 WS-GROUP-ALPHANUMERIC PIC X(5).
13          PROCEDURE DIVISION USING WS-GROUP.
B 14            DISPLAY WS-GROUP-ALPHABETIC.
15              CALL 'subsubsample' USING BY CONTENT WS-GROUP
16              END-CALL.
17          END PROGRAM subsample.
18
C:/GC-AWORK/subsample.cbl
breakpoint
```

now we type the F command again, then select the "sample.cbl" program and press Enter

```
Prompt dei comandi - q4
COBGDB                    GnuCOBOL GDB Interpreter
  1         IDENTIFICATION DIVISION.
  2         PROGRAM-ID. subsample.
  3         ENVIRONMENT DIVISION.
  4         DATA DIVISION.
  5         WORKING-STORAGE SECTION.
  6      Source Files
  7      C:/GC-AWORK/subsample.cbl
  8      C:/GC-AWORK/sample.cbl
  9      C:/GC-AWORK/subsubsample.cbl
  10
  11
  12
  13
B 14
  15
  16
  17
  18
C:/GC-AWORK/subsample.cbl
```

now we are back to the sample.cbl program to continue the debugging session as we need.

```
Prompt dei comandi - q4
COBGDB                    GnuCOBOL GDB Interpreter
  5         01 WS-NUMERIC PIC 9(2) VALUE 45.
  6         01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
  7         01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
  8         01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
  9         01 WS-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
  10        01 WS-GROUP.
  11              05 WS-GROUP-NUMERIC PIC 9(2) VALUE 45.
  12              05 WS-GROUP-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
  13              05 WS-GROUP-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
  14              05 WS-GROUP-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
  15              05 WS-GROUP-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
  16        01 WS-CHECK PIC 9(2).
  17              88 WS-CHECK-LITTLE VALUES ARE 50 THRU 99.
  18              88 WS-CHECK-BIG VALUES ARE 00 THRU 49.
  19        PROCEDURE DIVISION.
B>20            DISPLAY "Hello".
  21            CALL 'subsample' USING BY CONTENT WS-GROUP
  22            END-CALL.
  23            DISPLAY "World"
  24            STOP RUN.
C:/GC-AWORK/sample.cbl
Debugging
```

2.11. Run Command

If you click the Run command during a debug session you will receive a confirmation request, because Yes will restart a new the debugging session from first Procedure Division executable statement:

```
Prompt dei comandi - q3
COBGDB                    GnuCOBOL GDB Interpreter
 100                10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102        PROCEDURE DIVISION.
B 103        001-START.
 104            SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 105            SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
 106            SET ENVIRONMENT 'ESCDELAY' TO '25'
 107            *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 108        Message                                          > WS-STATU
 109        Would you like to "Run" the program again ? (= Restart)
>110                                  Yes    No
 111
 112            PERFORM UNTIL E-EXIT
 113                MOVE "MENU" TO WS-OP
 114                MOVE "CHOOSE AN OPTION" TO WS-STATUS   *> WS-STATUS
 115                MOVE SPACES TO WS-CHOICE                WS-NUMC012
 116                DISPLAY SS-CLS                                0
 117                ACCEPT SS-MENU
 118                                                          WS_NUMR
 119                EVALUATE TRUE                                25
 120                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
run
```

Note: if program has DECLARATIVES then the first automatic B Breakpoint will be settled at first executable PROCEDURE DIVISION statement that is the one after END DECLARATIVES, see following sample:

```
Prompt dei comandi - q1
COBGDB                    GnuCOBOL GDB Interpreter
 310            MOVE "ok" TO w-flag
 311        ELSE
 312            MOVE "N " TO w-flag
 313        END-IF
 314    ELSE
 315        DISPLAY "Error " w-fsRep " on REPORT FILE "
 316        STOP RUN
 317    END-IF.
 318 ex-err-Rep-x. EXIT.
 319 END DECLARATIVES.
 320
 321 *> ***********************************************************************
 322 *>
 323 *> ***********************************************************************
B>324 MAIN1 SECTION.
 325 MAIN-LOOP.
 326    perform InitialSettings thru InitialSettingsEx
 327
 328    PERFORM UNTIL wCRT-STATUS = K-ESCAPE
 329        DISPLAY screen-menu
 330        ACCEPT  screen-menu
C:/GC-AWORK/MASTER.cob
Debugging
```

2.12. Window Size command

When you start a debugging session, the screen size is 24 x 80 columns.

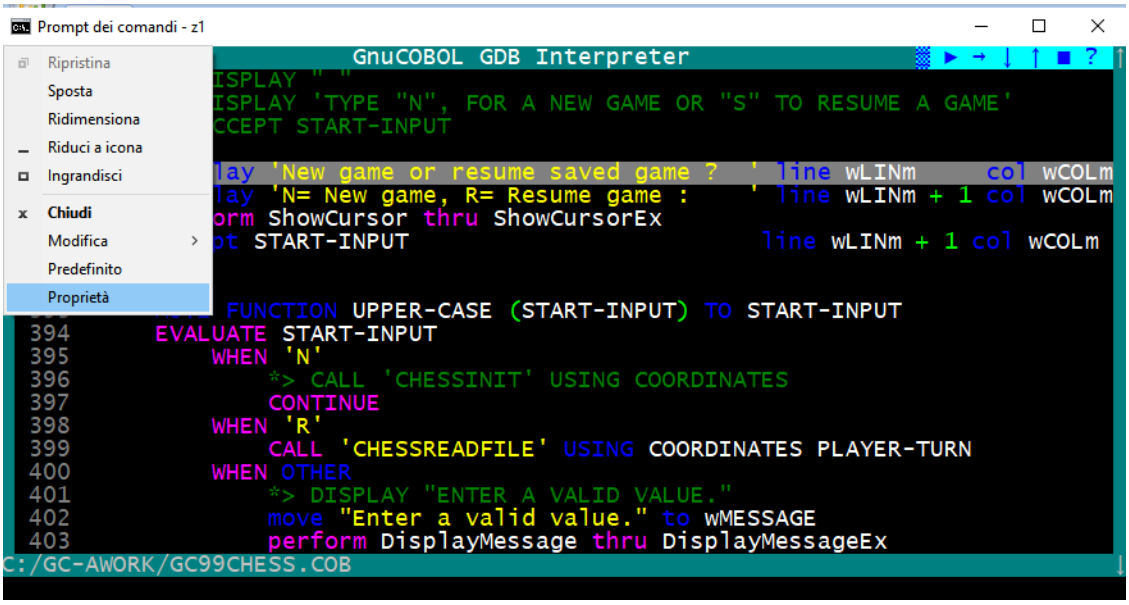
```
Prompt dei comandi - z1
COBGDB                    GnuCOBOL GDB Interpreter
 703        MOVE 1 TO U-CNT
 704
 705        *> EVALUATE WHITE
 706        EVALUATE TRUE
 707            WHEN PLAYER-TURN = 'w'
>708                PERFORM UNTIL INP-X-VAR-N = W-X-VAR(U-CNT)
 709                          AND INP-Y-VAR   = W-Y-POS(U-CNT)
 710                          OR  U-CNT         >= 16
 711
 712                          *> DEBUG                                  INP-Y-VAR
 713                          *> display INP-X-VAR-N     at 2260            "7"
 714                          *> display W-X-VAR(U-CNT) at 2265
 715                          *> display INP-Y-VAR       at  WS-COUNTERS
 716                          *> display W-Y-POS(U-CNT) at  "917040000000000"
 717                          *> DISPLAY U-CNT           at
 718            COORDINATES
 719            "051508WKY041208WQY010308WRY082408WRY020608WHY072108WHY03090
 720
 721                    ADD 1 TO U-CNT                                  INP-X-VAR-N
 722                END-PERFORM                                             "8"
 723                EVALUATE TRUE
C:/GC-AWORK/GC99CHESS.COB
Debugging
```

Type **W** (Window Size) command to switch between two window size of the debugger : 24 x 80 or 34 x 132.

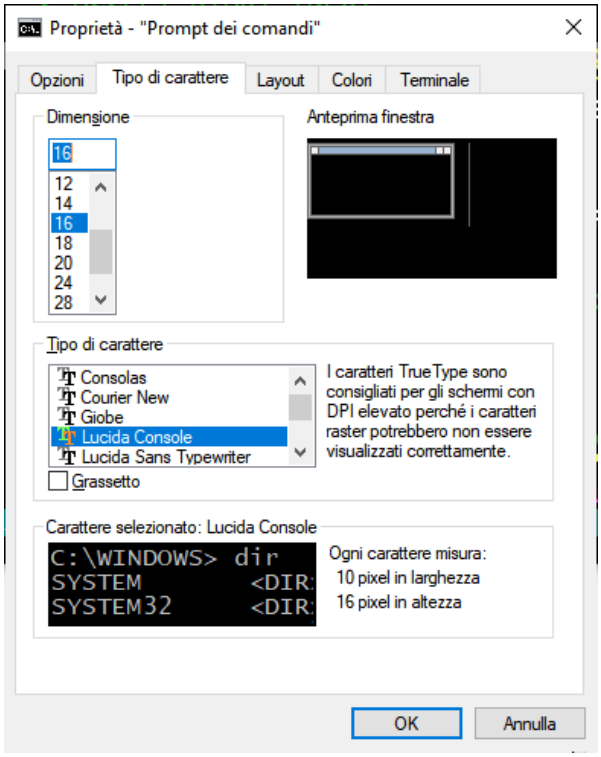
```
Prompt dei comandi - z1
COBGDB                    GnuCOBOL GDB Interpreter
 703        MOVE 1 TO U-CNT
 704
 705        *> EVALUATE WHITE
 706        EVALUATE TRUE
 707            WHEN PLAYER-TURN = 'w'
>708                PERFORM UNTIL INP-X-VAR-N = W-X-VAR(U-CNT)
 709                          AND INP-Y-VAR   = W-Y-POS(U-CNT)
 710                          OR  U-CNT         >= 16
 711
 712                          *> DEBUG
 713                          *> display INP-X-VAR-N     at 2260                INP-Y-VAR
 714                          *> display W-X-VAR(U-CNT) at 2265                     "7"
 715                          *> display INP-Y-VAR       at 2360
 716                          *> display W-Y-POS(U-CNT) at 2365   COORDINATES
 717                          *> DISPLAY U-CNT           at 2460  "051508WKY041208WQY010308WRY082408WRY020608WHY072108WHY03090
 718                          *> accept omitted
 719                          *> END DEBUG                                          INP-X-VAR-N
 720                                                                                     "8"
 721                    ADD 1 TO U-CNT
 722                END-PERFORM                                                   WS-COUNTERS
 723                EVALUATE TRUE                                                "917040000000000"
 724                    WHEN W-PIECE(U-CNT) = '--'
 725                        SET NOT-VALID TO TRUE                              WS-COUNTERS.U-CNT
 726                        *> DISPLAY "NO VALID PIECE HERE"                            4
 727                        move "No valid piece here." to wMESSAGE
 728                        perform DisplayMessage thru DisplayMessageEx
 729                        PERFORM GA-ACCEPT-CORD
 730                    WHEN INP-Y-VAR = W-Y-POS(U-CNT)
 731                        *> DISPLAY "YOU HAVE SELECTED " W-PIECE(U-CNT) " IN POSITION " INP-X-VAR INP-Y-VAR
 732                        string "Selecterd " W-PIECE(U-CNT) " in pos. " INP-X-VAR INP-Y-VAR into wMESSAGE
 733                        perform DisplayMessage thru DisplayMessageEx
C:/GC-AWORK/GC99CHESS.COB
Debugging
```

This can be very useful when you write GnuCOBOL code with the GnuCOBOL FREE FORMAT source option where each line of code can be longer than the classic 80 characters of GnuCOBOL FIXED FORMAT.

Warning.
Make sure the system font size is not too large otherwise the W command cannot display the 132 columns.
For a Windows environment use "Property" menu item to chek or change the Font size:



and then select a suitable font, for example Lucida Console of 16 might be adequate.



2.13. Quit Command

To close the debug session use the **Q** Quit command or left click with mouse the ■ button

```
Prompt dei comandi
  102         PROCEDURE DIVISION.
B 103         001-START.
  104             SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
  105             SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
  106             SET ENVIRONMENT 'ESCDELAY' TO '25'
  107             *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
  108             *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
  109             ACCEPT WS_NUMR FROM LINES
 >110             ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
  111             PERFORM 007-OPEN-FILES
  112             PERFORM UNTIL E-EXIT
  113                 MOVE "MENU" TO WS-OP
  114                 MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
  115                 MOVE SPACES TO WS-CHOICE                 WS-NUMC012
  116                 DISPLAY SS-CLS                                  0
  117                 ACCEPT SS-MENU
  118
  119                 EVALUATE TRUE
  120                     WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
quit
The end of the COBGDB execution.

C:\GC-AWORK>
```

3. Other Line Commands

3.1. Debugging a pre-compiled Program

You can use COBGDB to debug a previously generated executable file ex. **`prog.exe`**.
To do this, you must first compile the GnuCOBOL program **`prog.cob`** with these options:

```
cobc -g -fsource-location -ftraceall -v -O0 -x prog.cob prog2.cob ...
```

To start debugging without recompile the program, run cobgdb using the **`--exe`** directive as follows:

Windows:

```
cobgdb --exe prog.exe
```

Linux:

```
cobgdb --exe prog
```

3.2. COBGDB Version

Use command option: **`cobgdb --version`**

to display COBGDB version informations as follows:

```
COBGDB - GnuCobol GDB Interpreter - version 1.1.0
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
This COBGDB was configured as "MinGW32".
For bug reporting instructions, please see:
<https://github.com/marcsosduma/cobgdb>.
The end of the COBGDB execution.
```

4. Document Change Log

CHANGE LOG
Version 1 of 2023.12.12. *First release* . ***Version 2 of 2023.12.23.*** *Step by Step sample of use is added* *Some minor changes* . ***Version 3 of 2024.02.18.*** *Restructured showing new cobgdb screens and features* . ***Version 4 of 2024.04.01 and 20240403.*** *Added EDIT subcommand at H Show Command when viewing the variable from a line of code* *Added cobgdb --version option* . ***Version 5 of 2024.05.01.*** *Added the W Window size command to change User interface screen size* . ***Version 6 of 2024.05.05.*** *Added the --exe option to debug precompiled program* . ***Version 7 of 2025.04.24.*** *Added the new D command useful to enable or disable automatic display variables during debugging / animation*

Technical info