DOCUMENT CODE	MODULE: xxxxxxxxxx	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	1	

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.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	4	

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 .

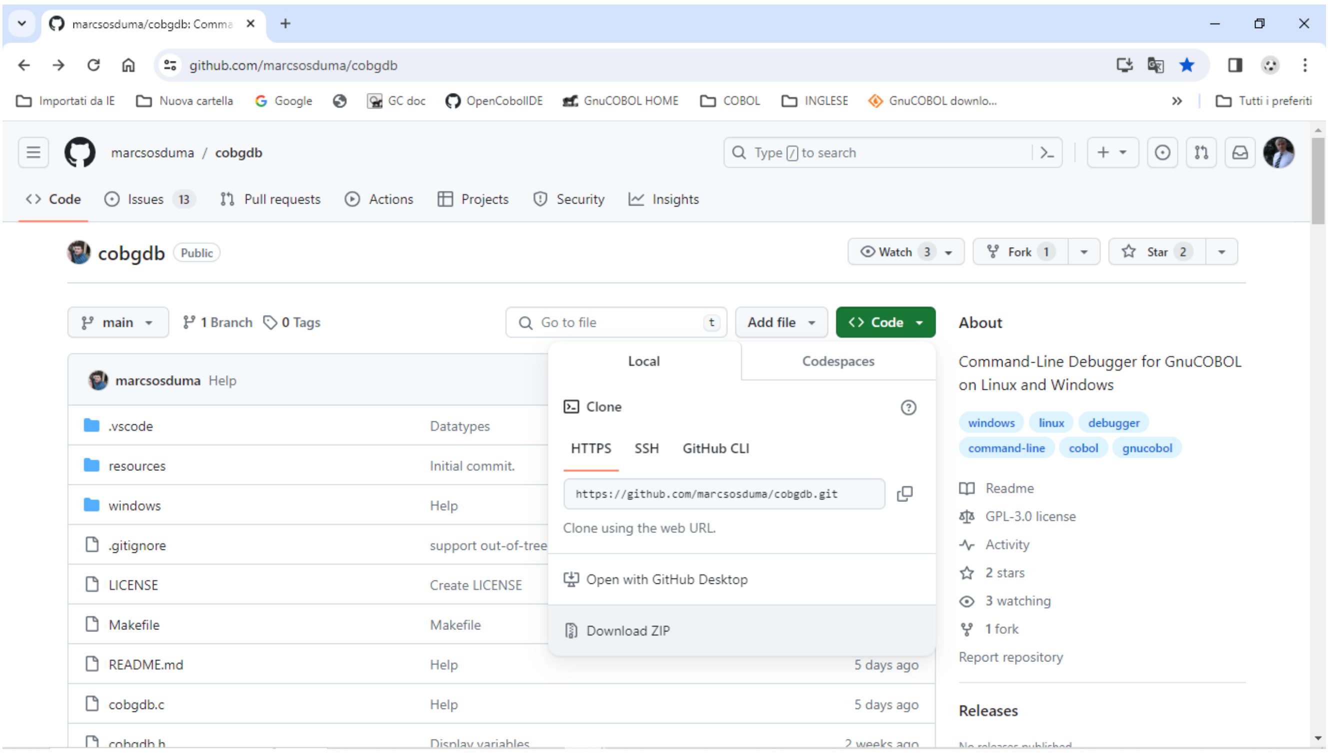
As an example you can put `cobgdb.exe` into the "`bin`" folder of your GnuCOBOL installation (the same folder where the GnuCOBOL compiler `cobc.exe` is located)

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 the `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 to debug.
To debug sub programs separately , see also "Debugging sub Programs" chapter in this document.

You can also 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.
O	fOcus	When program encounters an ACCEPT verb, this command switch the focus from debugging / animation screen to the application screen to accept an user input and then automatic go back to the debugging / animation screen.
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.
CTRL-F	Find	Search for text in the source code
CTRL-L	Go to	Go to Line.
Q	Quit	Quits (ends) the debugging / animation session and the program (or programs).

2. Tutorial - Sample Debugging Session

This tutorial is on a Windows 10 platform using following version of GnuCOBOL:

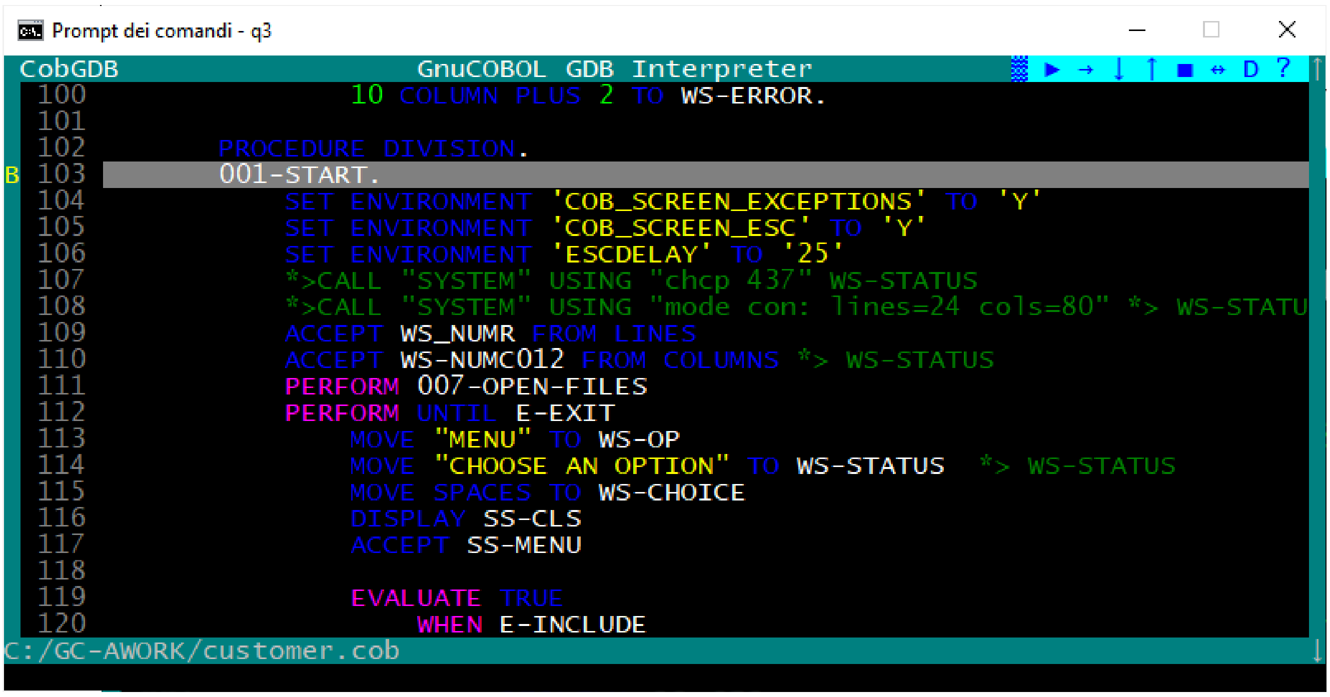
```
cobc (GnuCOBOL) 3.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
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).

DOCUMENT CODE	MODULE: xxxxxxxxxx	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	8	

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:



```
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
```

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):

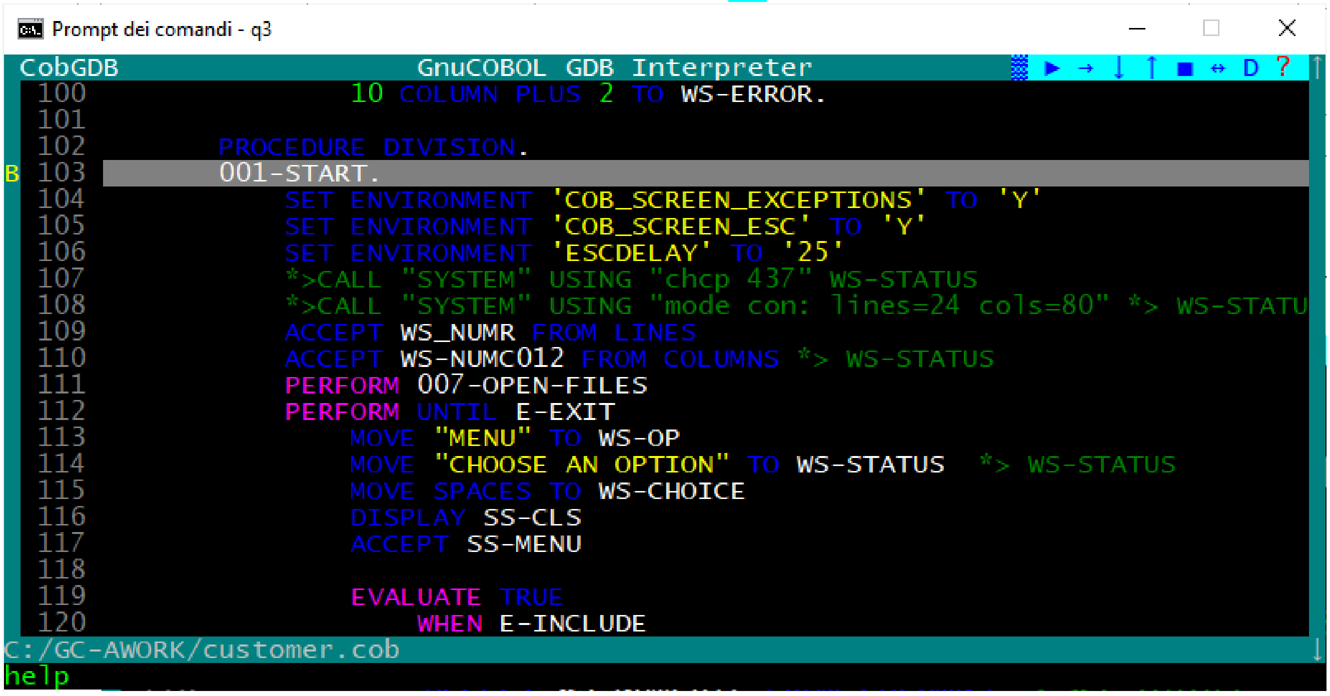
> → ↓ ↑ ■ <-> D ?

>	=	**Run** command
→	=	**Next** command
↓	=	**Step** command
↑	=	**Go** command
■	=	**Quit** command
<->	=	**Focus** 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.

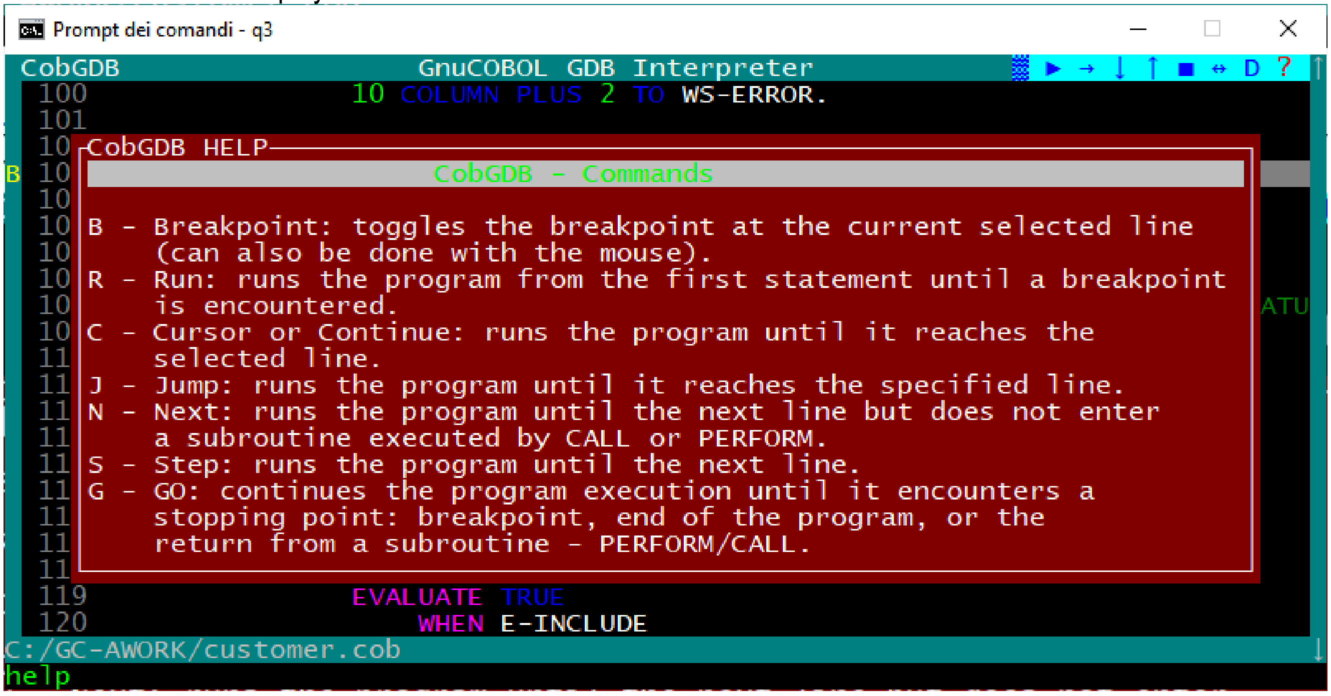
2.1. Help Command

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



```
Prompt dei comandi - q3
CobGDB                   GnuCOBOL GDB Interpreter
                    10 COLUMN PLUS 2 TO WS-ERROR.

           PROCEDURE DIVISION.
B          001-START.
               SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
               SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
               SET ENVIRONMENT 'ESCDELAY' TO '25'
               *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
               *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
               ACCEPT WS_NUMR FROM LINES
               ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
               PERFORM 007-OPEN-FILES
               PERFORM UNTIL E-EXIT
                   MOVE "MENU" TO WS-OP
                   MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
                   MOVE SPACES TO WS-CHOICE
                   DISPLAY SS-CLS
                   ACCEPT SS-MENU

                   EVALUATE TRUE
                       WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
help
```

the HELP window is displayed



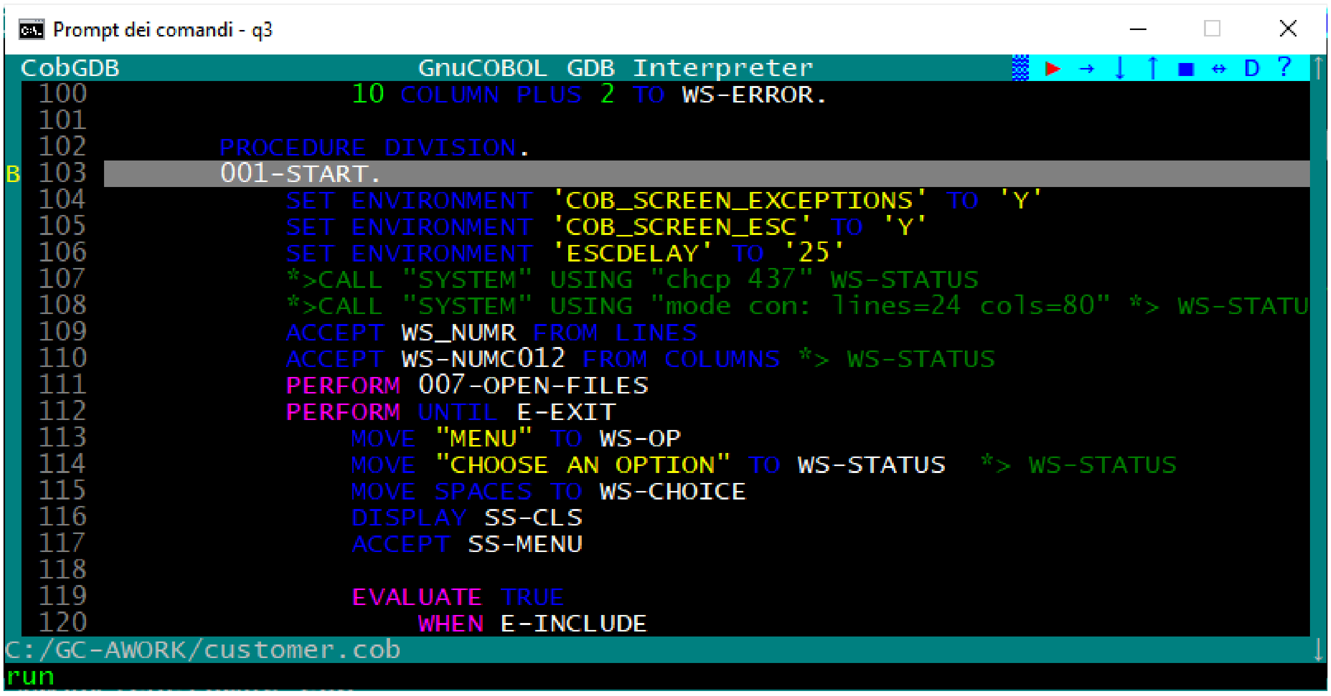
```
Prompt dei comandi - q3
CobGDB                   GnuCOBOL GDB Interpreter
                    10 COLUMN PLUS 2 TO WS-ERROR.

 ┌CobGDB HELP──────────────────────────────────────────────────┐
B│                    CobGDB - Commands                        │
 │                                                             │
 │B - Breakpoint: toggles the breakpoint at the current selected line
 │    (can also be done with the mouse).                       │
 │R - Run: runs the program from the first statement until a breakpoint
 │    is encountered.                                          │
 │C - Cursor or Continue: runs the program until it reaches the│
 │    selected line.                                           │
 │J - Jump: runs the program until it reaches the specified line.
 │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.             │
 │G - GO: continues the program execution until it encounters a│
 │    stopping point: breakpoint, end of the program, or the   │
 │    return from a subroutine - PERFORM/CALL.                 │
 └─────────────────────────────────────────────────────────────┘
                   EVALUATE TRUE
                       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.

DOCUMENT CODE	MODULE: xxxxxxxxxx	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	10	

2.2. Run Command

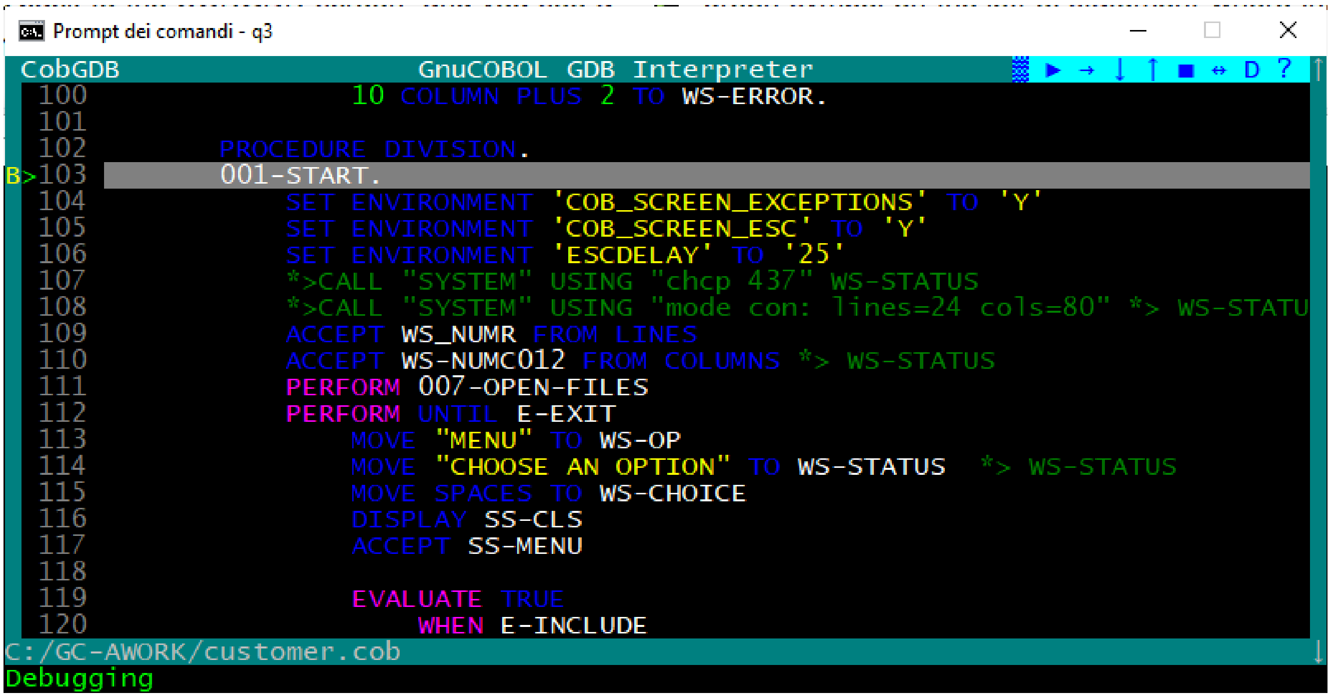
To start executing the program and the debugging session from first program statement you always must use the "**R**" command (key) or left click with mouse on the **>** **Run** button



```
CobGDB                          GnuCOBOL  GDB  Interpreter
                          10 COLUMN PLUS 2 TO WS-ERROR.

                  PROCEDURE DIVISION.
B                 001-START.
                      SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
                      SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
                      SET ENVIRONMENT 'ESCDELAY' TO '25'
                      *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
                      *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
                      ACCEPT WS_NUMR FROM LINES
                      ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
                      PERFORM 007-OPEN-FILES
                      PERFORM UNTIL E-EXIT
                          MOVE "MENU" TO WS-OP
                          MOVE "CHOOSE AN OPTION" TO WS-STATUS   *> WS-STATUS
                          MOVE SPACES TO WS-CHOICE
                          DISPLAY SS-CLS
                          ACCEPT SS-MENU

                          EVALUATE TRUE
                              WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
run
```

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

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):



```
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.

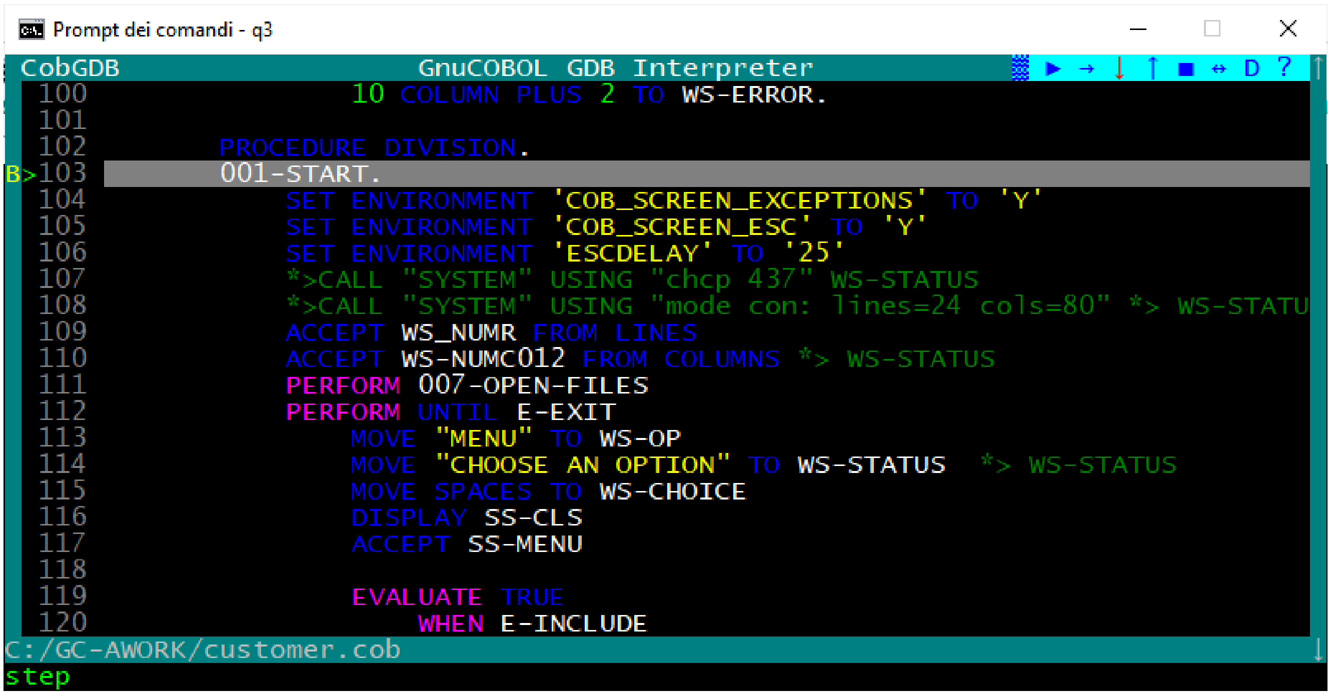
After pressing D button and during the source code animation, the debugger automatically shows some pop-up windows with variables content from the line in execution.

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.

DOCUMENT CODE	MODULE: XXXXXXXXXX	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	12	

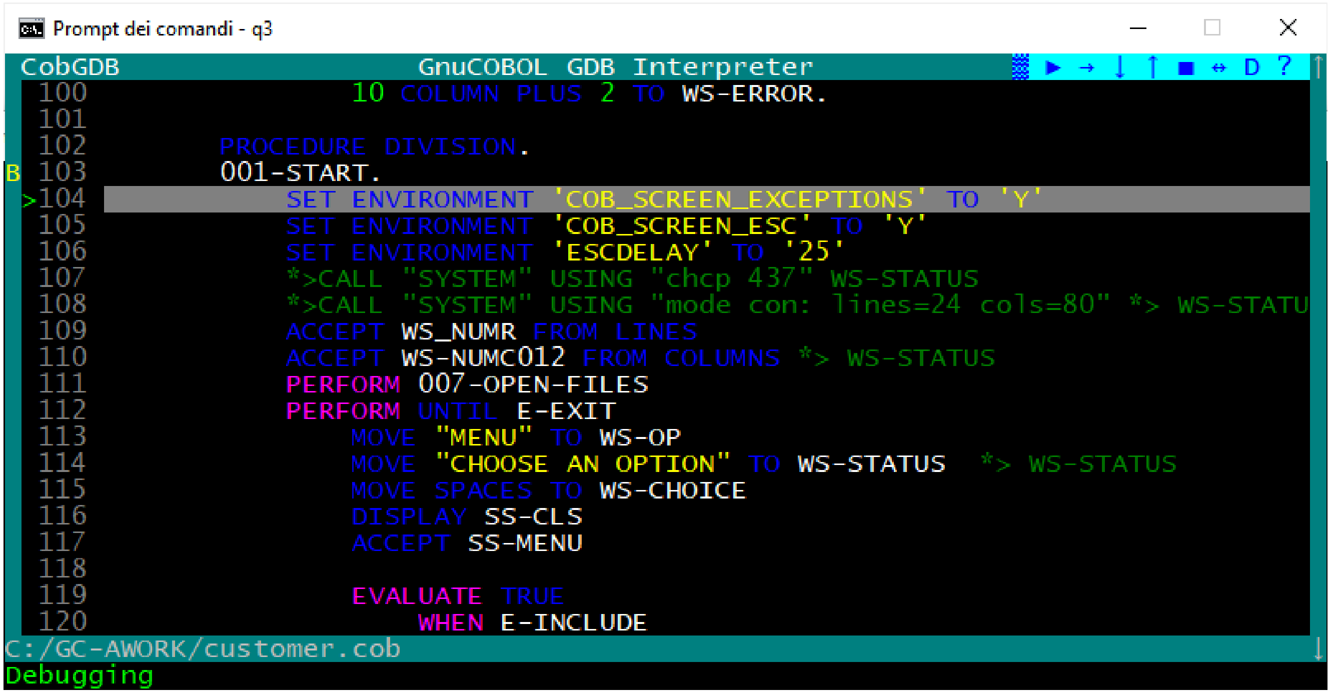
2.3. Step Command

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



```
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
step
```

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



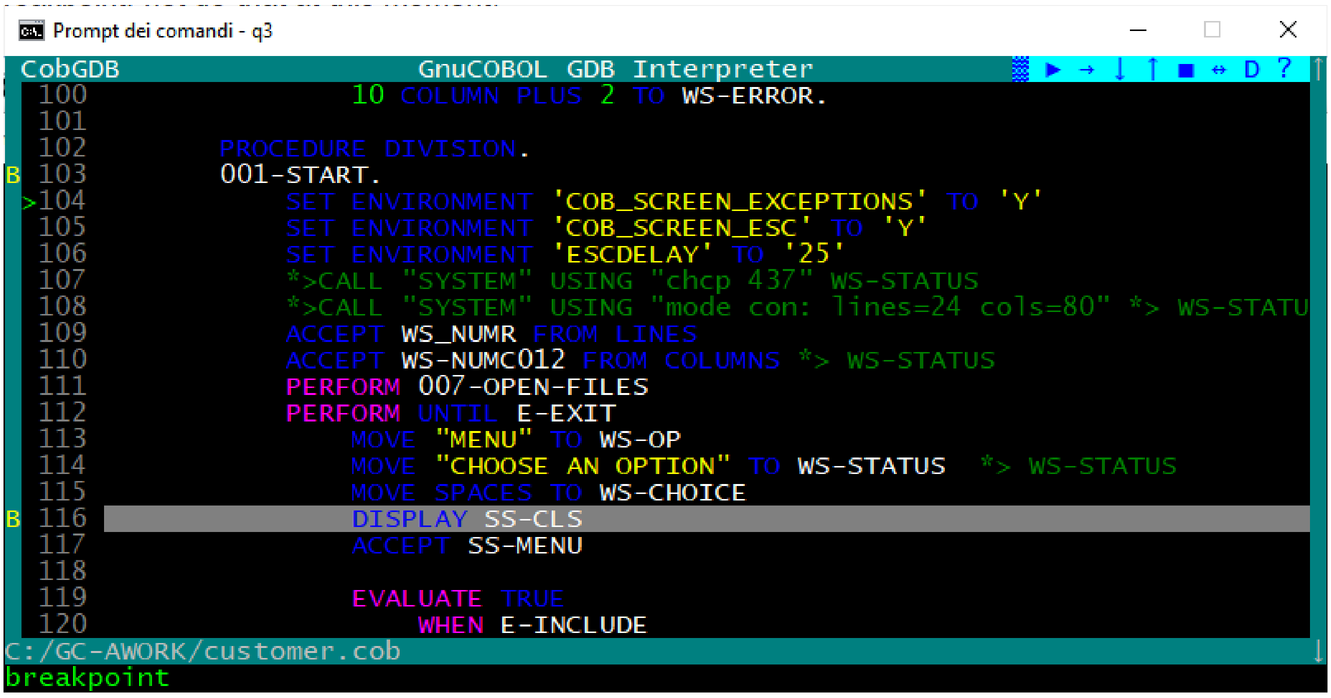
```
Prompt dei comandi - q3
CobGDB                       GnuCOBOL  GDB  Interpreter          ▓ ▶ → ↓ ↑ ■ ↔ D ?
 100                    10 COLUMN PLUS 2 TO WS-ERROR.
 101
 102            PROCEDURE DIVISION.
B 103            001-START.
 >104                SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
 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
Debugging
```

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

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	13	

- 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)

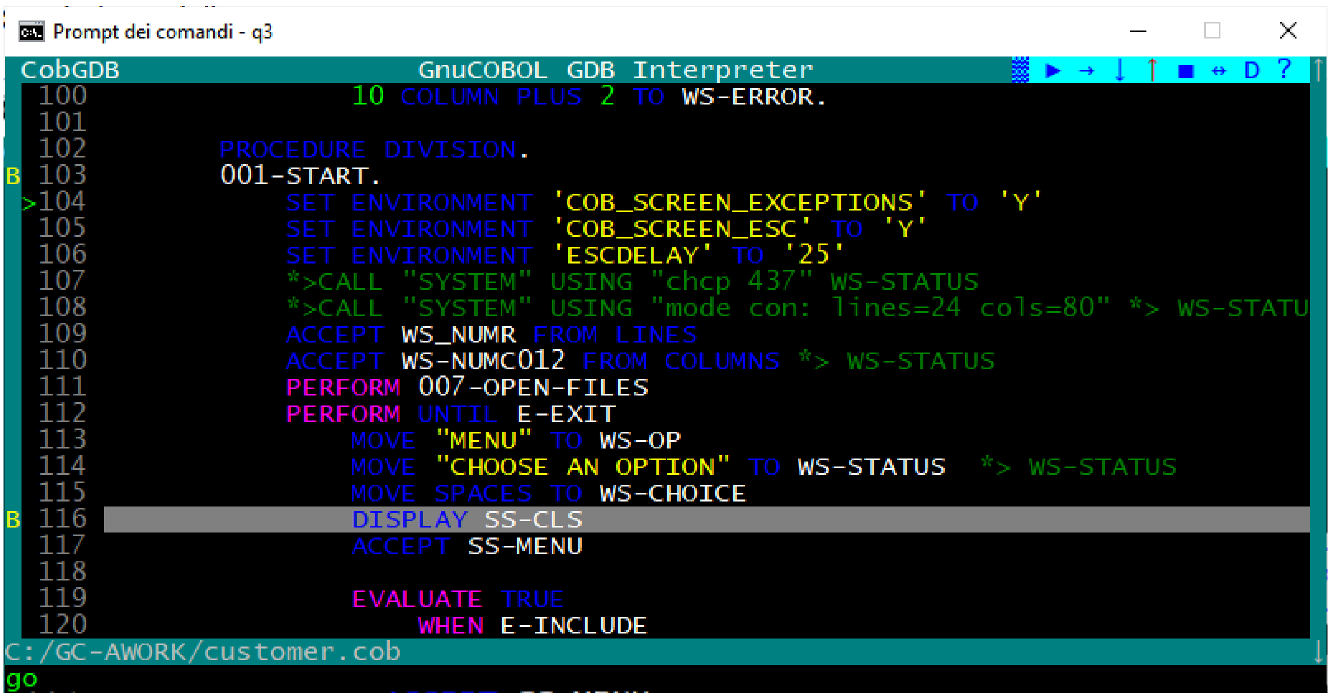
- The application displays a "**B**" on the left of the line (type B again - or re-click - when you want to delete the Breakpoint, not do that at this moment)



```
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
B 116                   DISPLAY SS-CLS
  117                   ACCEPT SS-MENU
  118
  119                   EVALUATE TRUE
  120                       WHEN E-INCLUDE
C:/GC-AWORK/customer.cob                                                              ↓
breakpoint
```

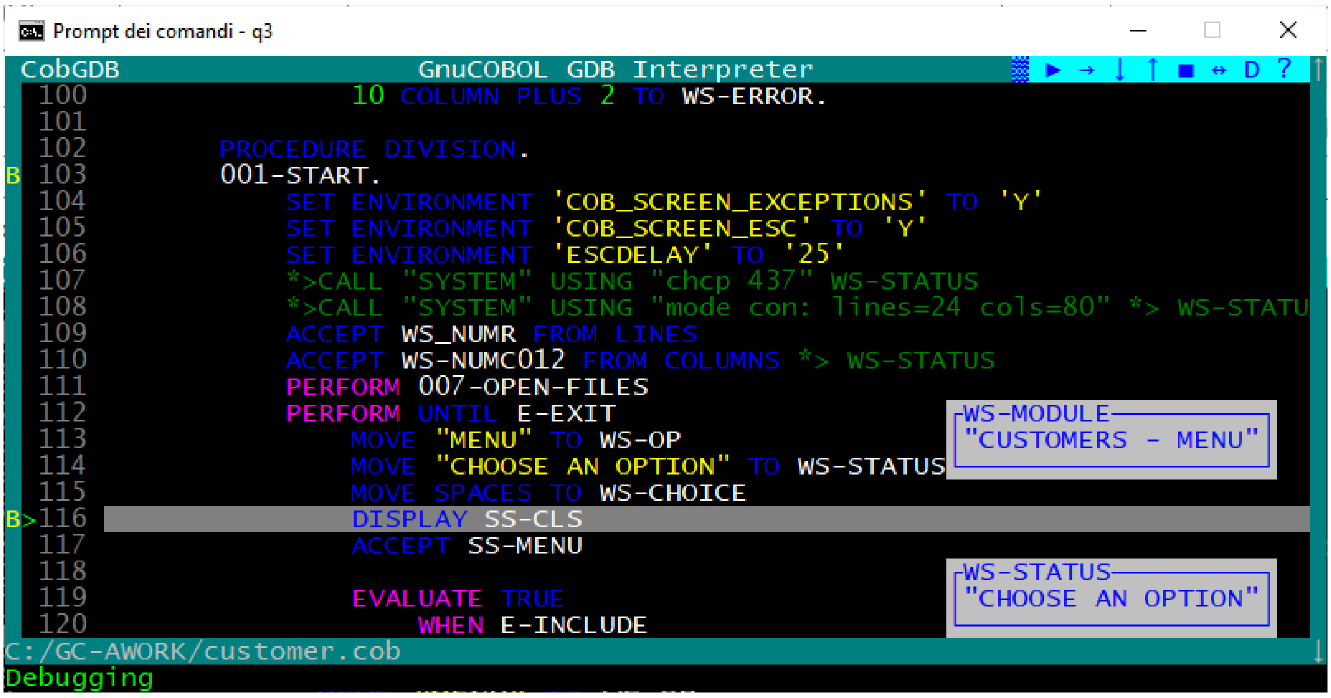
2.4. Go Command

Type **G** (Go) or left click with mouse the ↑ button to execute the program until a B Breakpoint is detected:



```
 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
go
```

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:

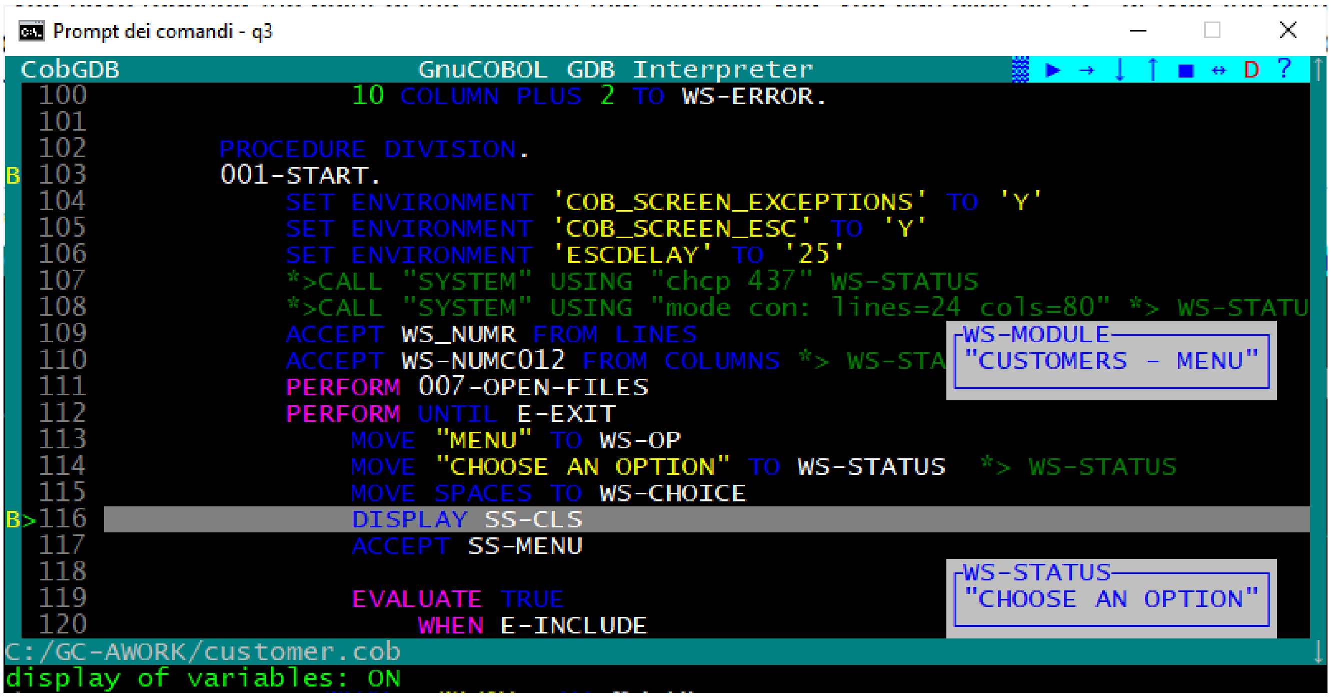


```
 CobGDB                   GnuCOBOL  GDB Interpreter
  100                   10 COLUMN PLUS 2 TO WS-ERROR.
  101
  102           PROCEDURE DIVISION.
B 103           001-START.
  104               SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
  105               SET ENVIRONMENT 'COB_SCREEN_ESC' TO '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-MODULE────────────┐
  113                   MOVE "MENU" TO WS-OP                  │"CUSTOMERS - MENU"   │
  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-STATUS────────────┐
  119                   EVALUATE TRUE                         │"CHOOSE AN OPTION"   │
  120                       WHEN E-INCLUDE                    └─────────────────────┘
C:/GC-AWORK/customer.cob
Debugging
```

DOCUMENT CODE	MODULE: xxxxxxxxxx	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	15	

2.5. Display Variables

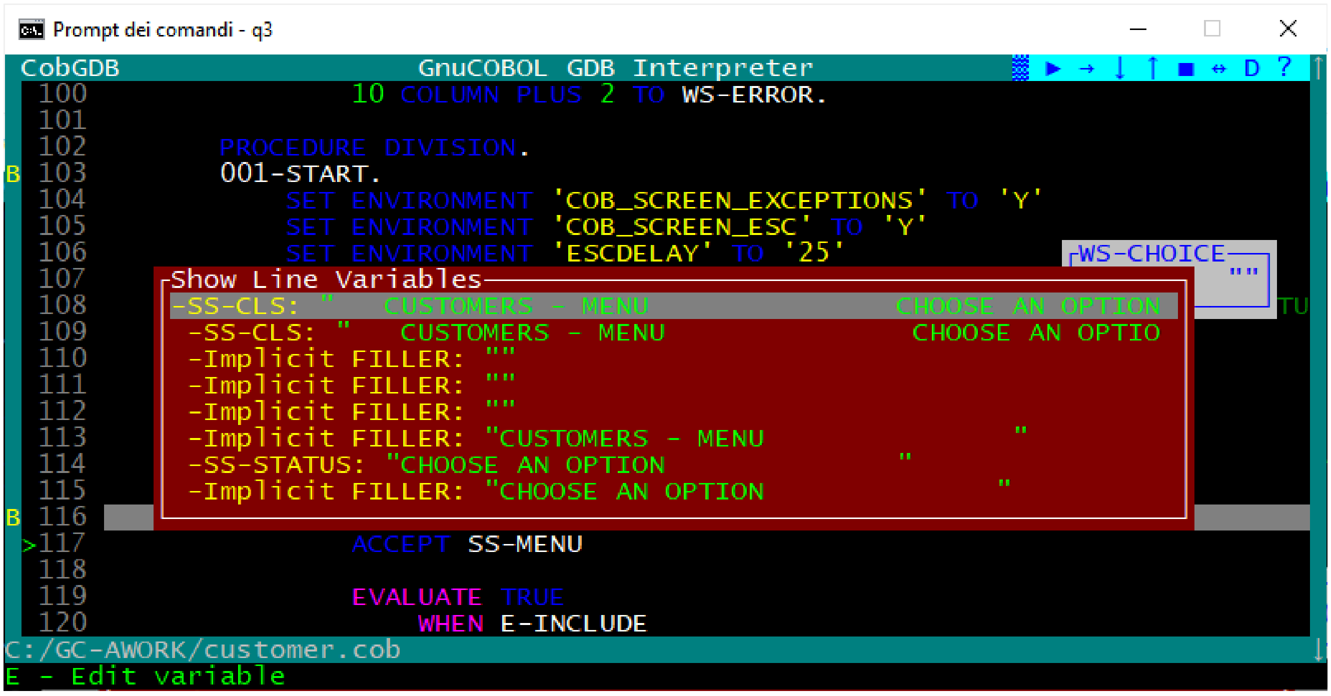
The debugger / animator can do an automatic display of all variables content from the statement executed and from the previous statement.
This can slow down the animation / execution and it may not be useful to have it activated everywhere.
Type **D** (Display Variables) command to enable or disable 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 pop-up windows) and of the previous statement (blue pop-up windows).



```
CobGDB                         GnuCOBOL  GDB Interpreter
                       10 COLUMN PLUS 2 TO WS-ERROR.

              PROCEDURE DIVISION.
B             001-START.
                  SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
                  SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
                  SET ENVIRONMENT 'ESCDELAY' TO '25'
                  *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
                  *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
                  ACCEPT WS_NUMR FROM LINES             ┌WS-MODULE─────────────┐
                  ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STA│"CUSTOMERS - MENU"    │
                  PERFORM 007-OPEN-FILES
                  PERFORM UNTIL E-EXIT
                      MOVE "MENU" TO WS-OP
                      MOVE "CHOOSE AN OPTION" TO WS-STATUS   *> WS-STATUS
                      MOVE SPACES TO WS-CHOICE
B>                    DISPLAY SS-CLS
                      ACCEPT SS-MENU
                                                        ┌WS-STATUS─────────────┐
                      EVALUATE TRUE                     │"CHOOSE AN OPTION"    │
                          WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
display of variables: ON
```

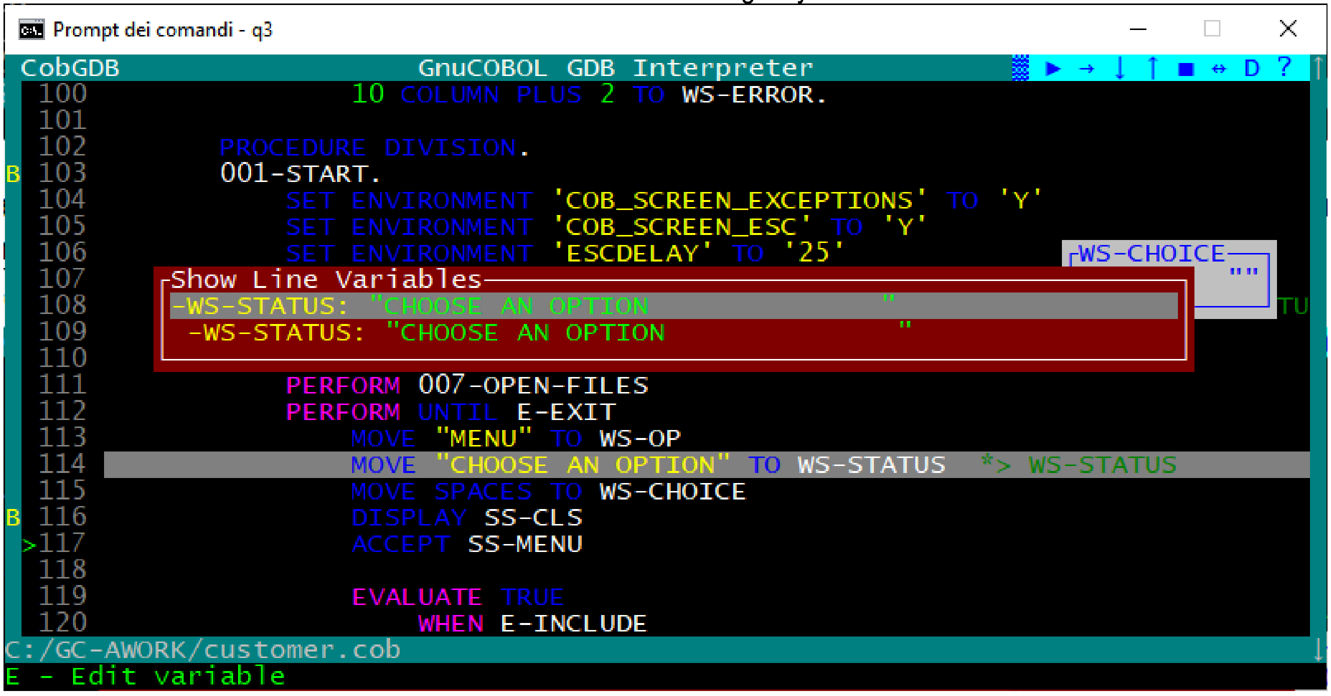
2.6. Show Command

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               ▓ ▶ → ↓ ↑ ■ ↔ 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'             ┌WS-CHOICE──┐
  107       ┌Show Line Variables──────────────────────────────────┐   ""     │
  108       │-SS-CLS: "    CUSTOMERS - MENU          CHOOSE AN OPTION│──────────┘TU
  109       │ -SS-CLS: "    CUSTOMERS - MENU          CHOOSE AN OPTIO│
  110       │ -Implicit FILLER: ""                                 │
  111       │ -Implicit FILLER: ""                                 │
  112       │ -Implicit FILLER: ""                                 │
  113       │ -Implicit FILLER: "CUSTOMERS - MENU          "       │
  114       │ -SS-STATUS: "CHOOSE AN OPTION          "             │
  115       │ -Implicit FILLER: "CHOOSE AN OPTION          "       │
B 116       └─────────────────────────────────────────────────────┘
 >117                   ACCEPT SS-MENU
  118
  119                   EVALUATE TRUE
  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:

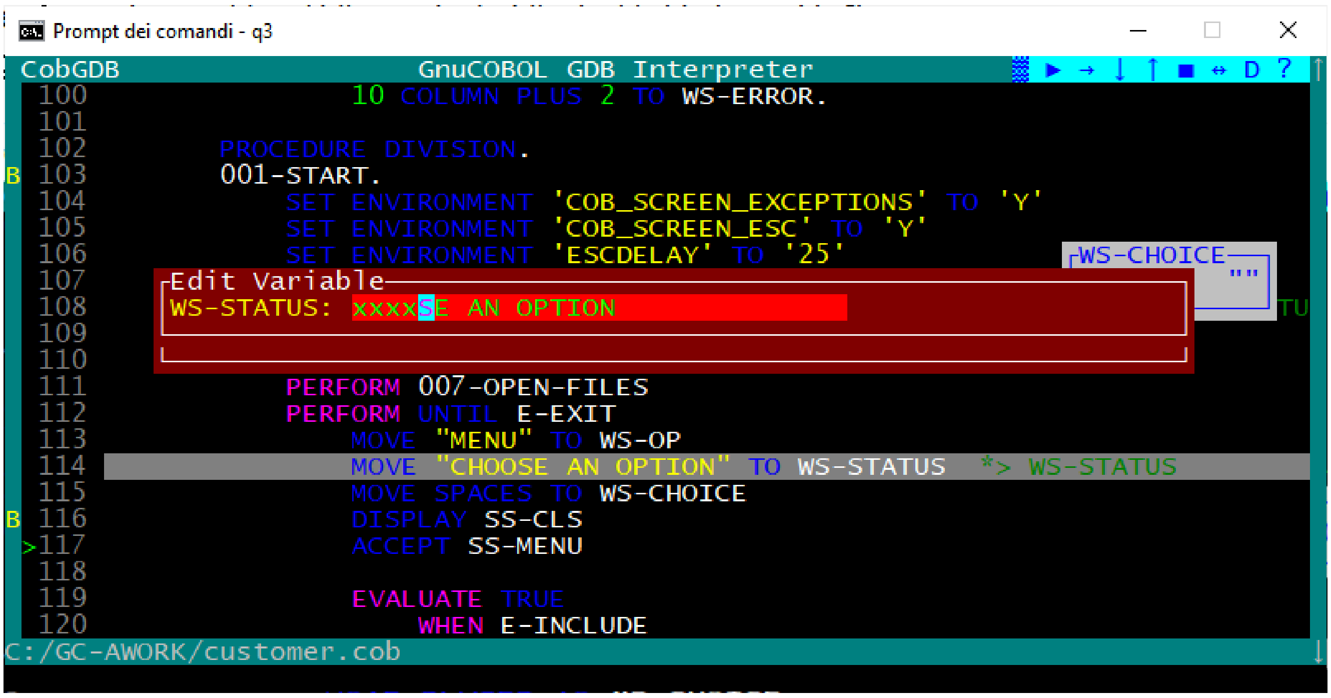


```
Prompt dei comandi - q3
CobGDB                  GnuCOBOL GDB Interpreter               ▓ ▶ → ↓ ↑ ■ ↔ D ?
  100                   10 COLUMN PLUS 2 TO WS-ERROR.
  101
  102           PROCEDURE DIVISION.
B 103           001-START.
  104               SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
  105               SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
  106               SET ENVIRONMENT 'ESCDELAY' TO '25'             ┌WS-CHOICE──┐
  107       ┌Show Line Variables──────────────────────────────────┐   ""     │
  108       │-WS-STATUS: "CHOOSE AN OPTION          "              │──────────┘TU
  109       │ -WS-STATUS: "CHOOSE AN OPTION          "             │
  110       └─────────────────────────────────────────────────────┘
  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
E - Edit variable
```

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

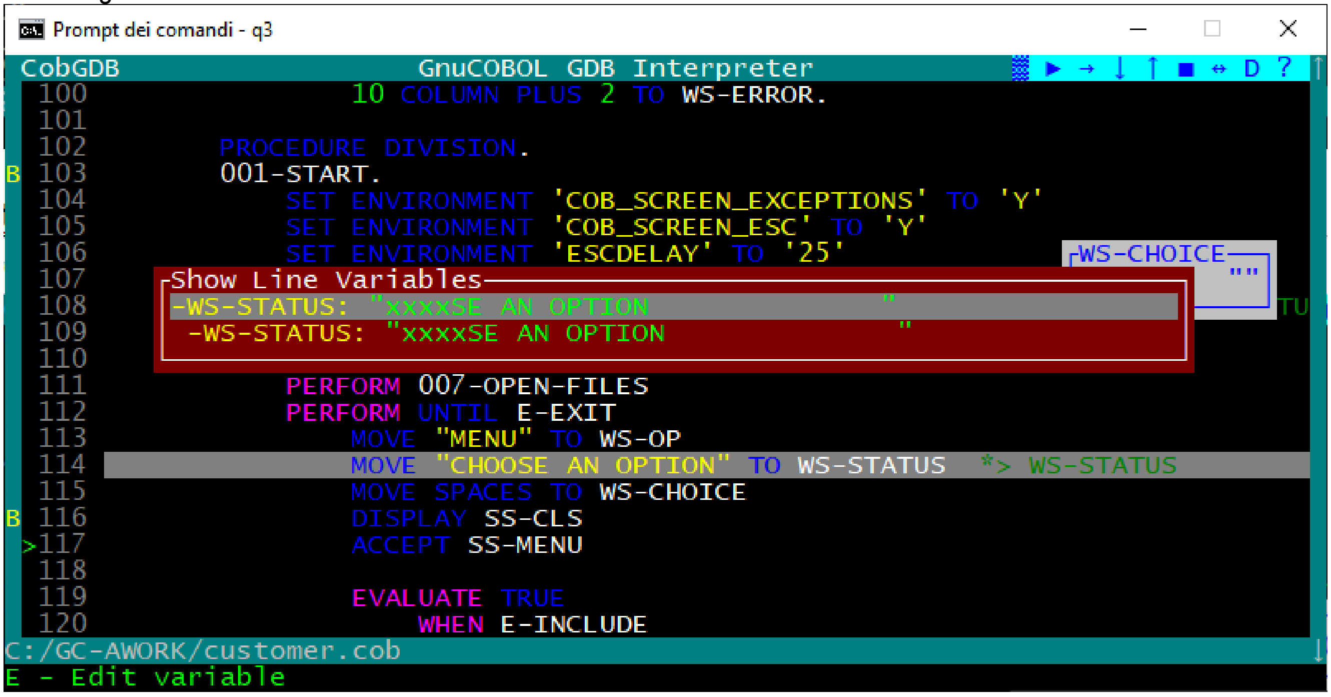
2.6.1. Edit subCommand

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                ▓ ► → ↓ ↑ ■ ↔ D ?
                        10 COLUMN PLUS 2 TO WS-ERROR.

                PROCEDURE DIVISION.
B               001-START.
                    SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
                    SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
                    SET ENVIRONMENT 'ESCDELAY' TO '25'        ┌WS-CHOICE──┐
          ┌Edit Variable─────────────────────────────────────────┐ │ ""  │
          │WS-STATUS: xxxxSE AN OPTION                           │ └─────┘TU
          │                                                      │
          └──────────────────────────────────────────────────────┘
                    PERFORM 007-OPEN-FILES
                    PERFORM UNTIL E-EXIT
                        MOVE "MENU" TO WS-OP
                        MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
                        MOVE SPACES TO WS-CHOICE
B                       DISPLAY SS-CLS
>                       ACCEPT SS-MENU

                        EVALUATE TRUE
                            WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
```

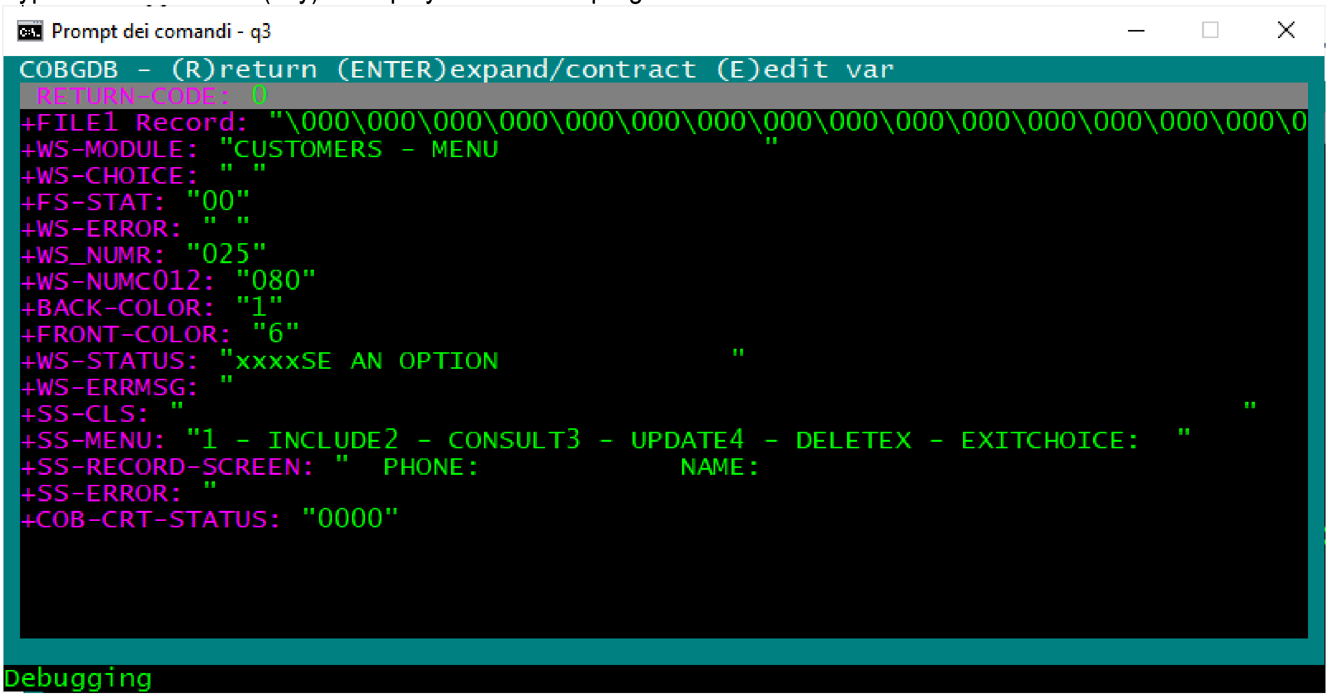
Resulting:



```
Prompt dei comandi - q3
CobGDB              GnuCOBOL  GDB  Interpreter                ▓ ► → ↓ ↑ ■ ↔ D ?
                        10 COLUMN PLUS 2 TO WS-ERROR.

                PROCEDURE DIVISION.
B               001-START.
                    SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
                    SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
                    SET ENVIRONMENT 'ESCDELAY' TO '25'        ┌WS-CHOICE──┐
          ┌Show Line Variables───────────────────────────────────┐ │ ""  │
          │-WS-STATUS: "xxxxSE AN OPTION                 "       │ └─────┘TU
          │ -WS-STATUS: "xxxxSE AN OPTION                 "      │
          └──────────────────────────────────────────────────────┘
                    PERFORM 007-OPEN-FILES
                    PERFORM UNTIL E-EXIT
                        MOVE "MENU" TO WS-OP
                        MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
                        MOVE SPACES TO WS-CHOICE
B                       DISPLAY SS-CLS
>                       ACCEPT SS-MENU

                        EVALUATE TRUE
                            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:

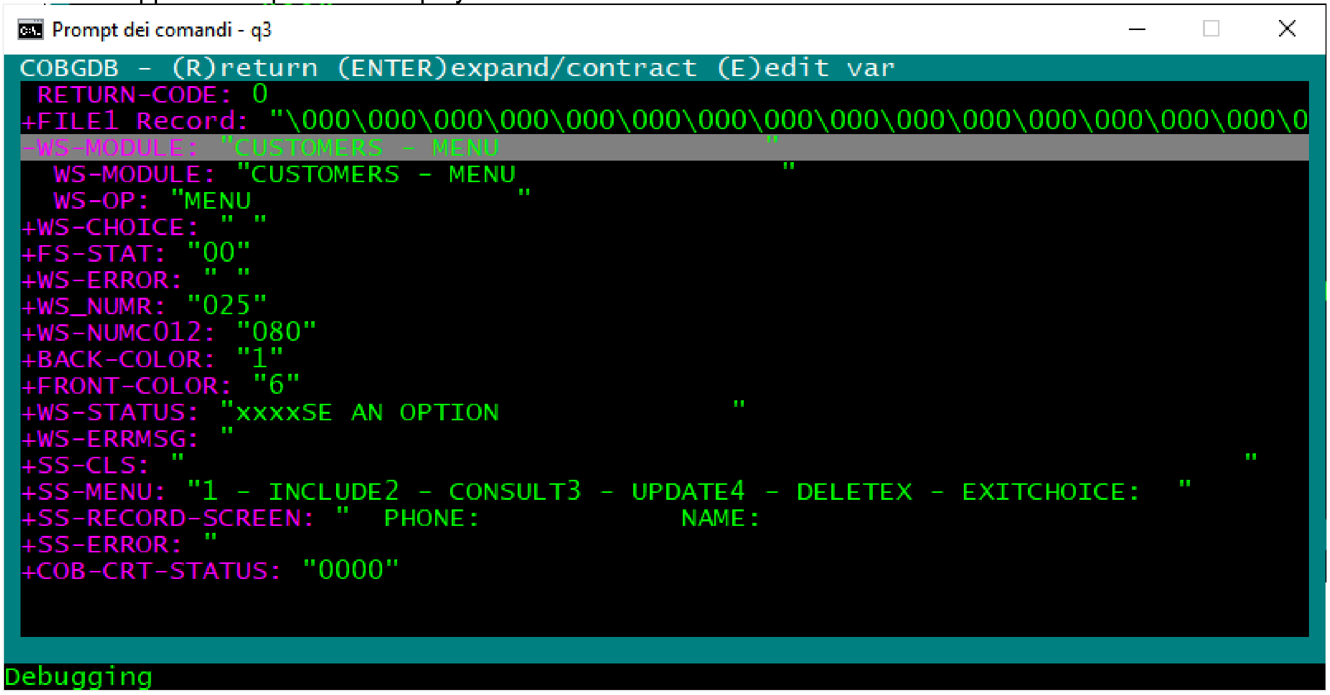


Prompt dei comandi - q3

```
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\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
```

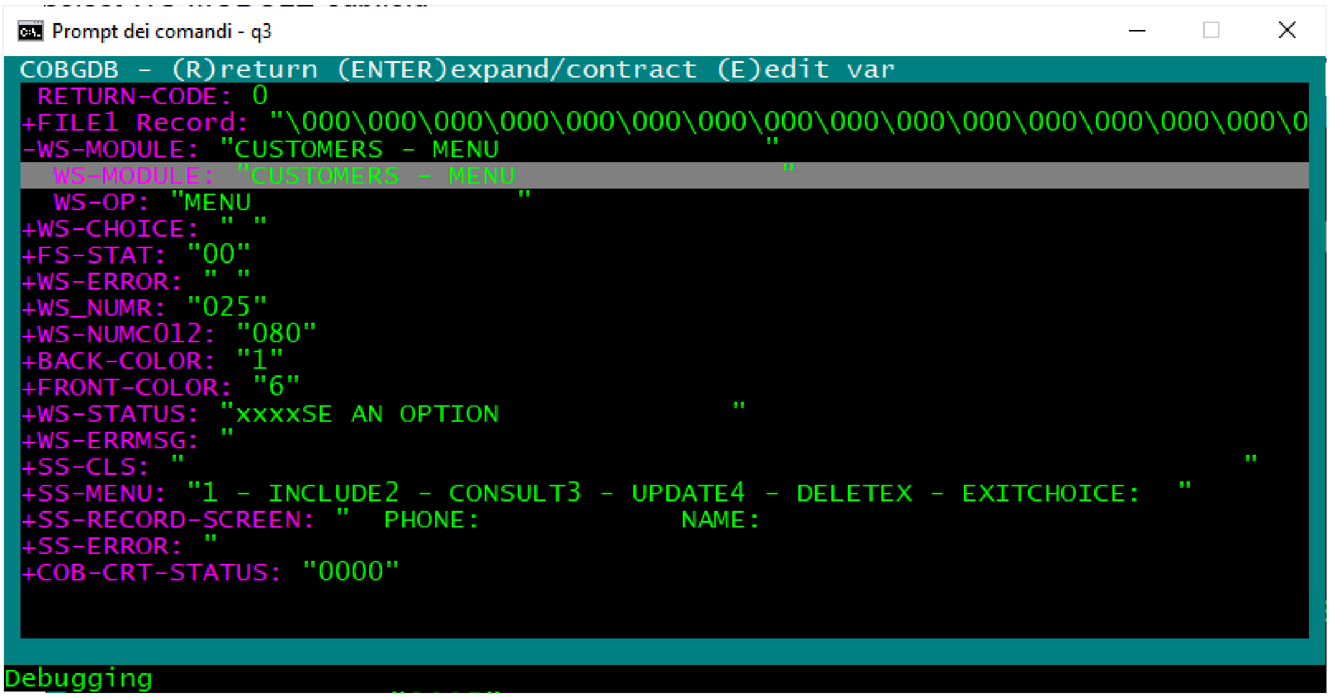
2.7.1.Enter subCommand

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\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

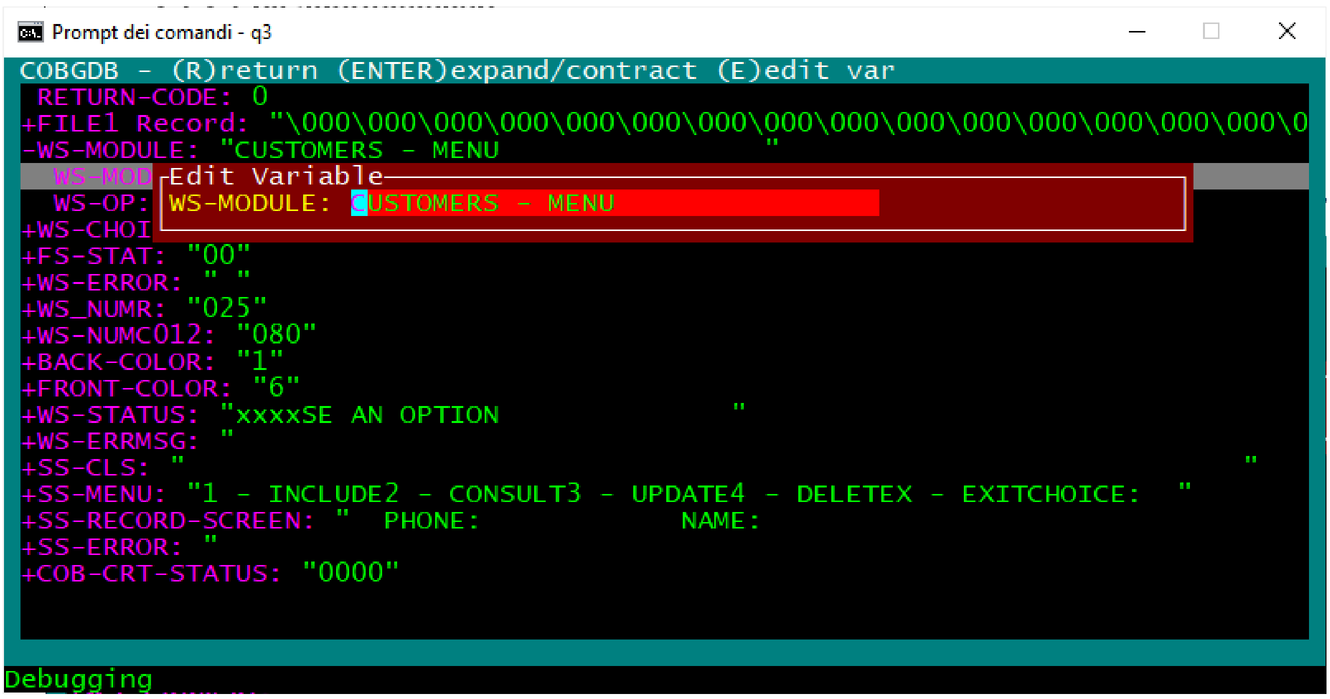


```
Prompt dei comandi - q3
COBGDB - (R)return (ENTER)expand/contract (E)edit var
 RETURN-CODE: 0
+FILE1 Record: "\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
-WS-MODULE: "CUSTOMERS - MENU                                "
  WS-MODULE: "CUSTOMERS - MENU                                "
  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
```

DOCUMENT CODE	MODULE: xxxxxxxxxx	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	20	

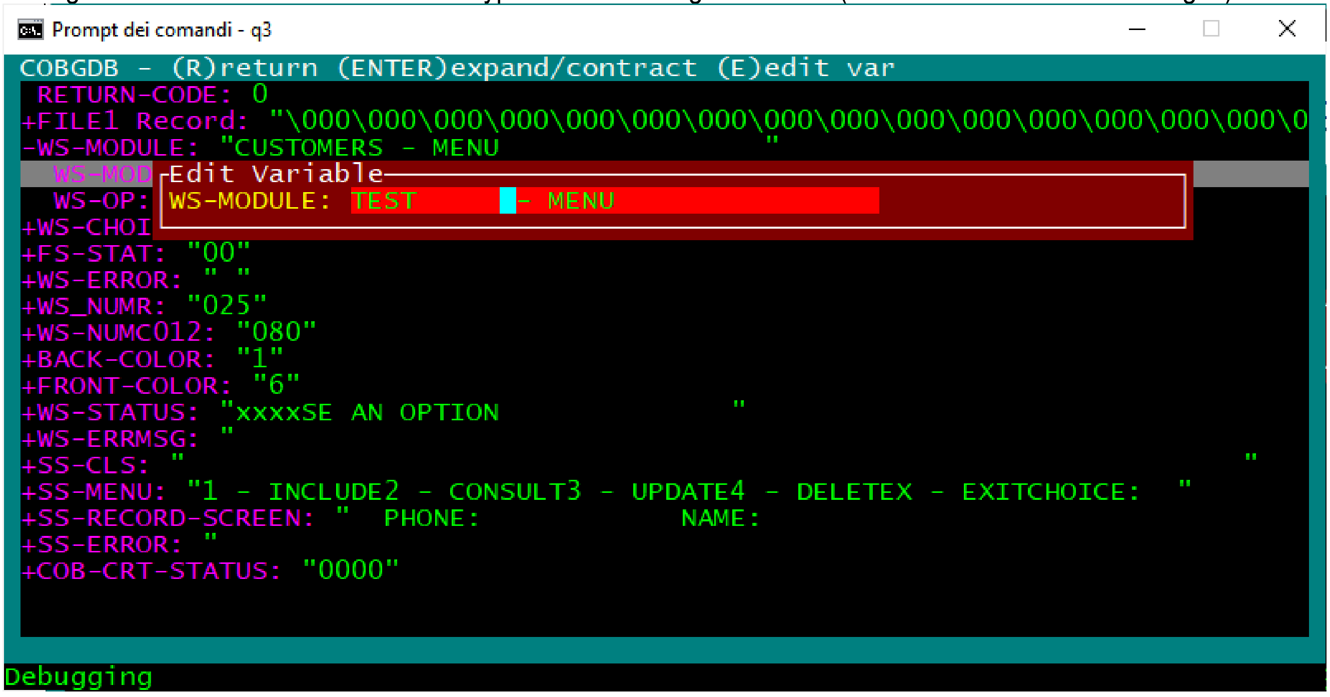
2.7.2. Edit subCommand

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) :



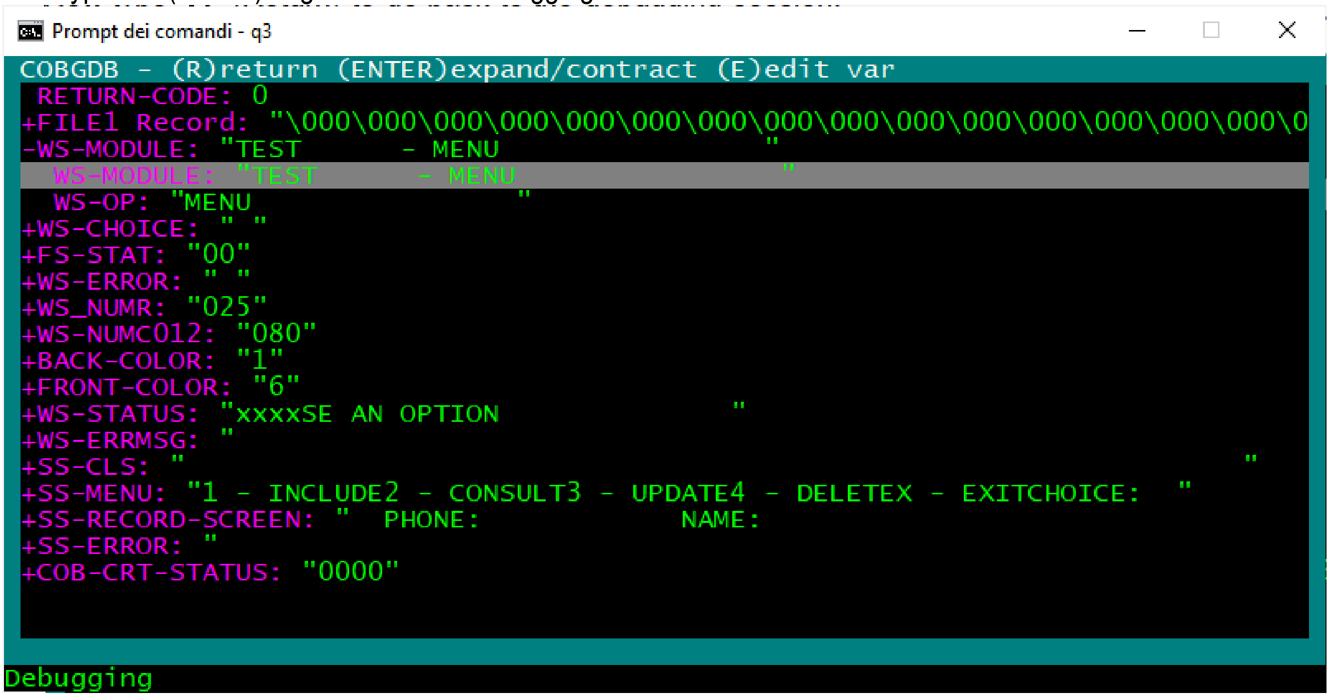
```
Prompt dei comandi - q3
COBGDB - (R)return (ENTER)expand/contract (E)edit var
 RETURN-CODE: 0
+FILE1 Record: "\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\0
-WS-MODULE: "CUSTOMERS - MENU                         "
  WS-MOD┌Edit Variable─────────────────────────────────────────────┐
  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
```

DOCUMENT CODE	MODULE: xxxxxxxxxx	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	21	

2.7.3. Return subCommand

WS-Module has new value.

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

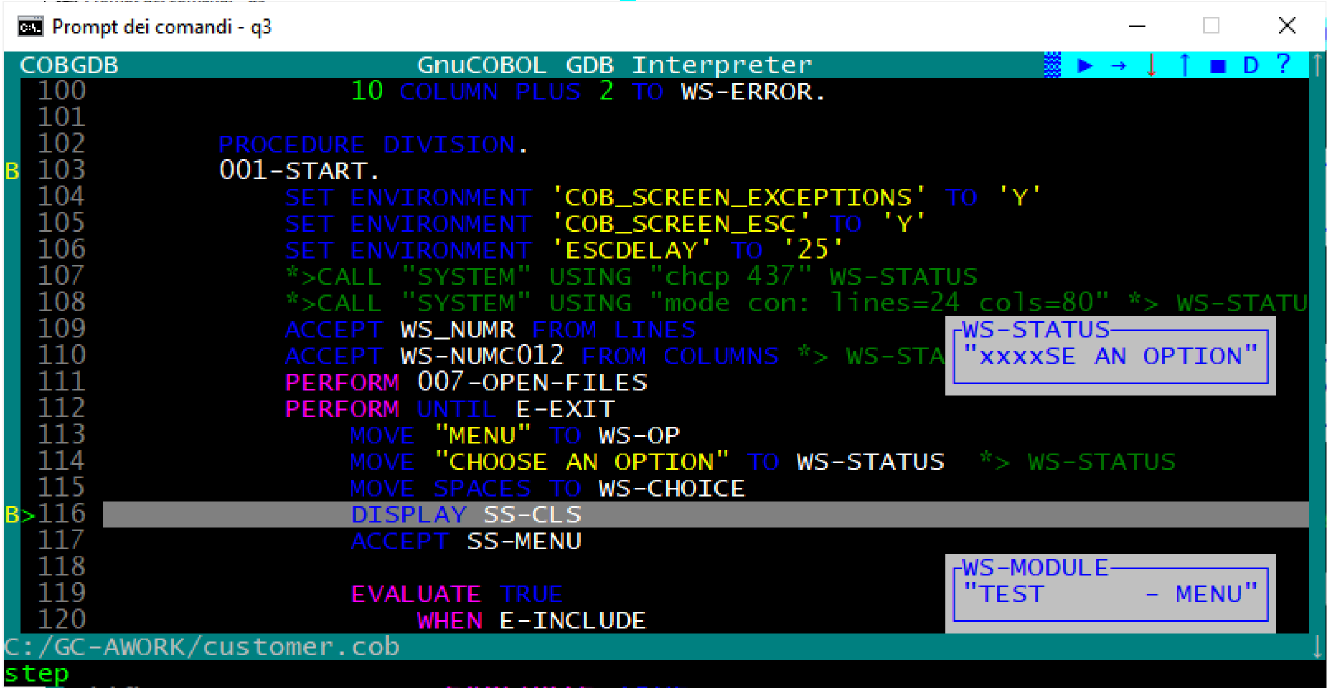

Prompt dei comandi - q3

```
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
```

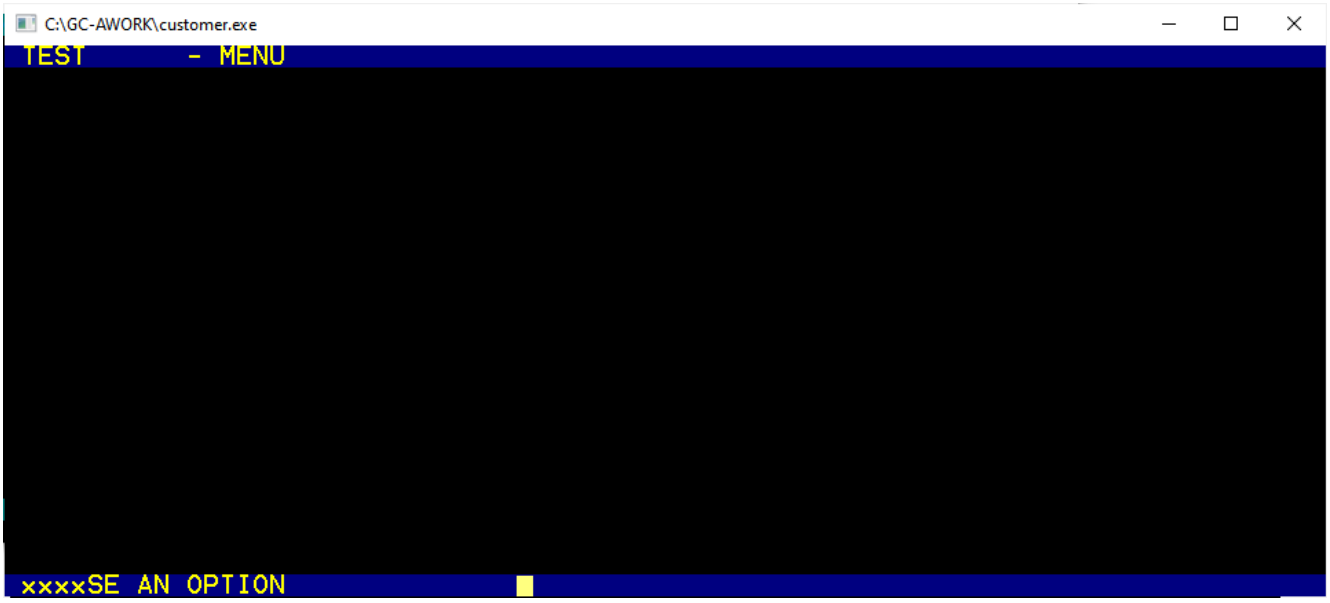
2.8. Step Command

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

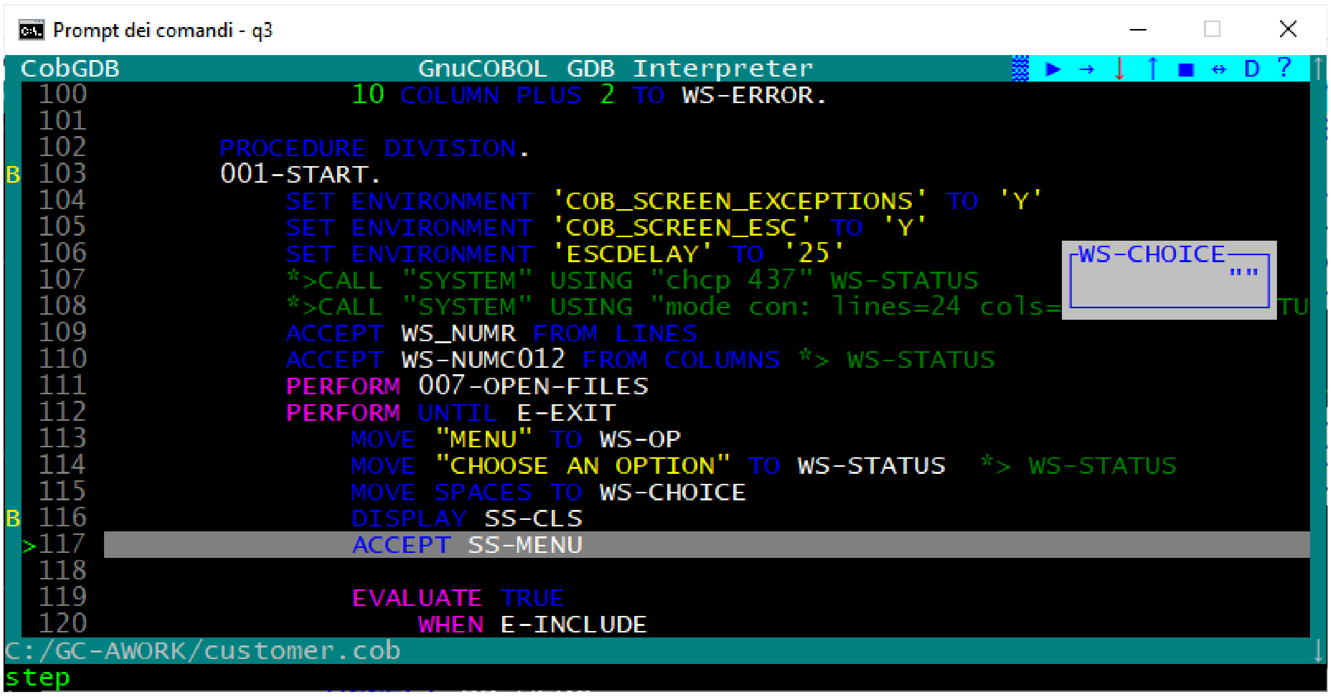


```
C:\GC-AWORK\customer.exe
 TEST      - MENU




 xxxxSE AN OPTION
```

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

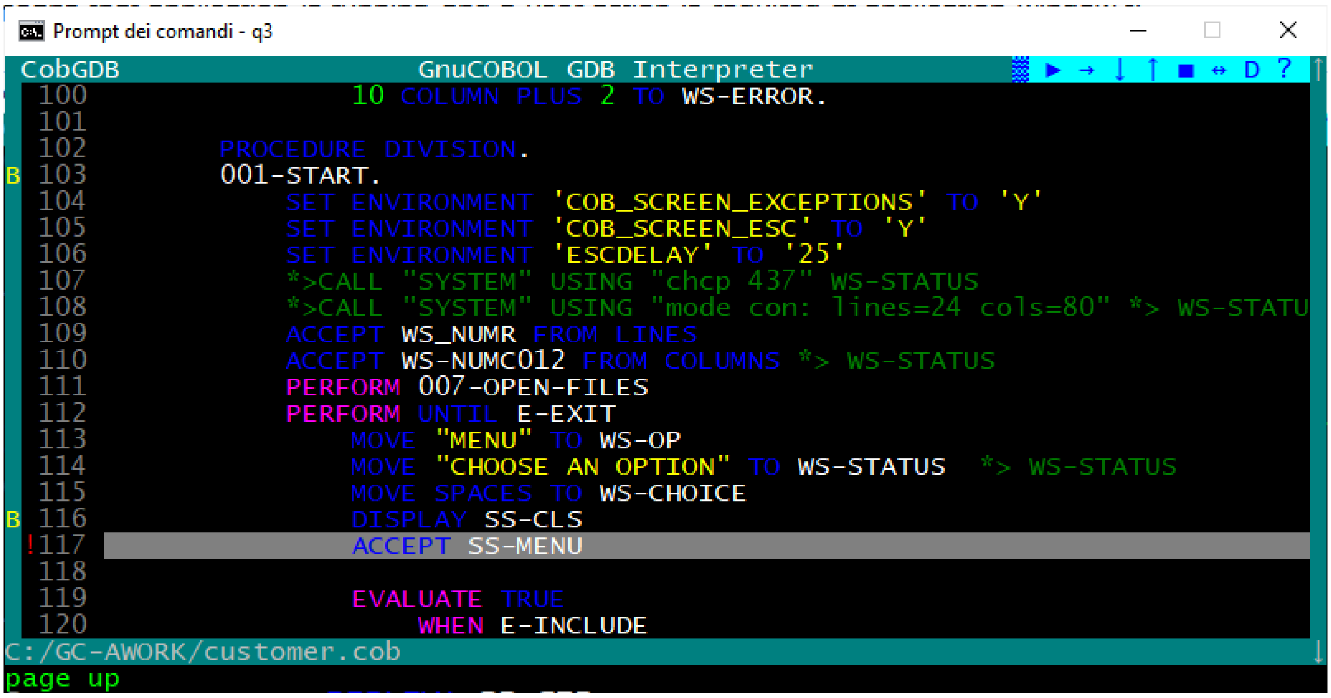
DOCUMENT CODE	MODULE: xxxxxxxxxx	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	23	

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                ▓ ▶ → ↓ ↑ ■ ↔ 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'            ┌WS-CHOICE──────┐
  107               *>CALL "SYSTEM" USING "chcp 437" WS-STATUS    │           ""  │
  108               *>CALL "SYSTEM" USING "mode con: lines=24 cols=└───────────────┘TU
  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
step
```

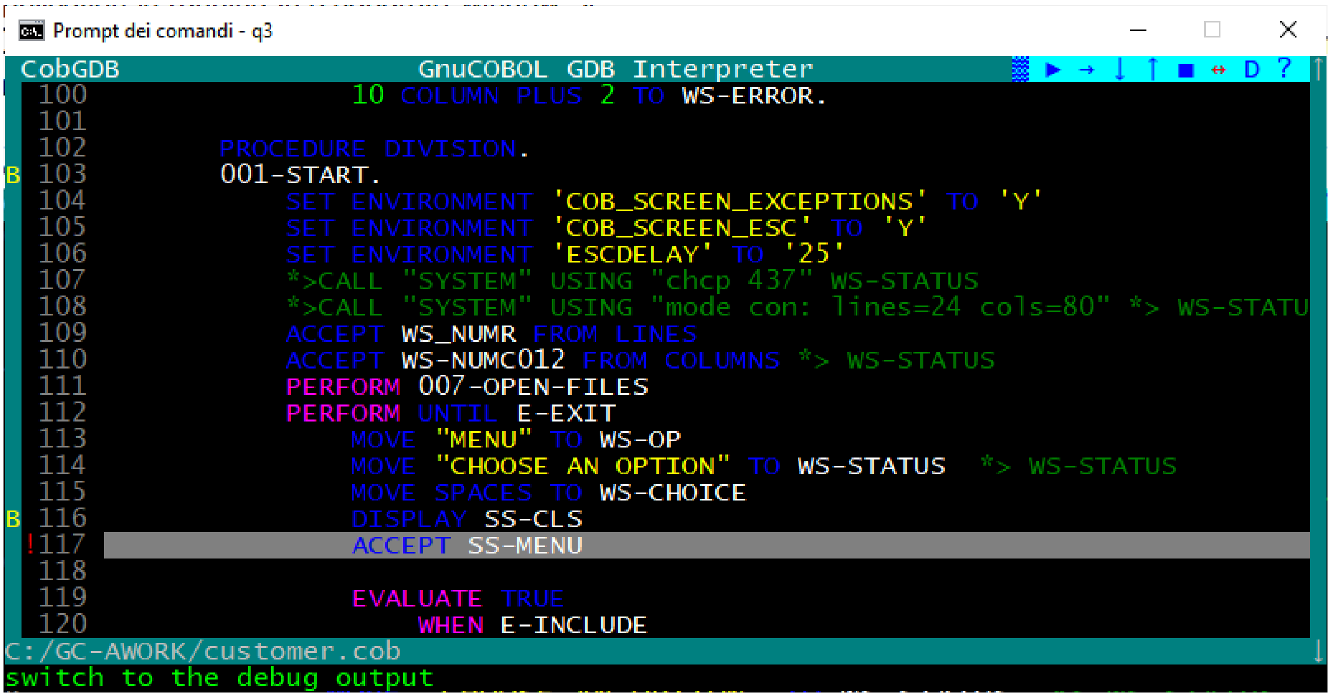
A red ! exclamation mark appears on the line !117 .
This means that the application is running and a user action is required at application window.



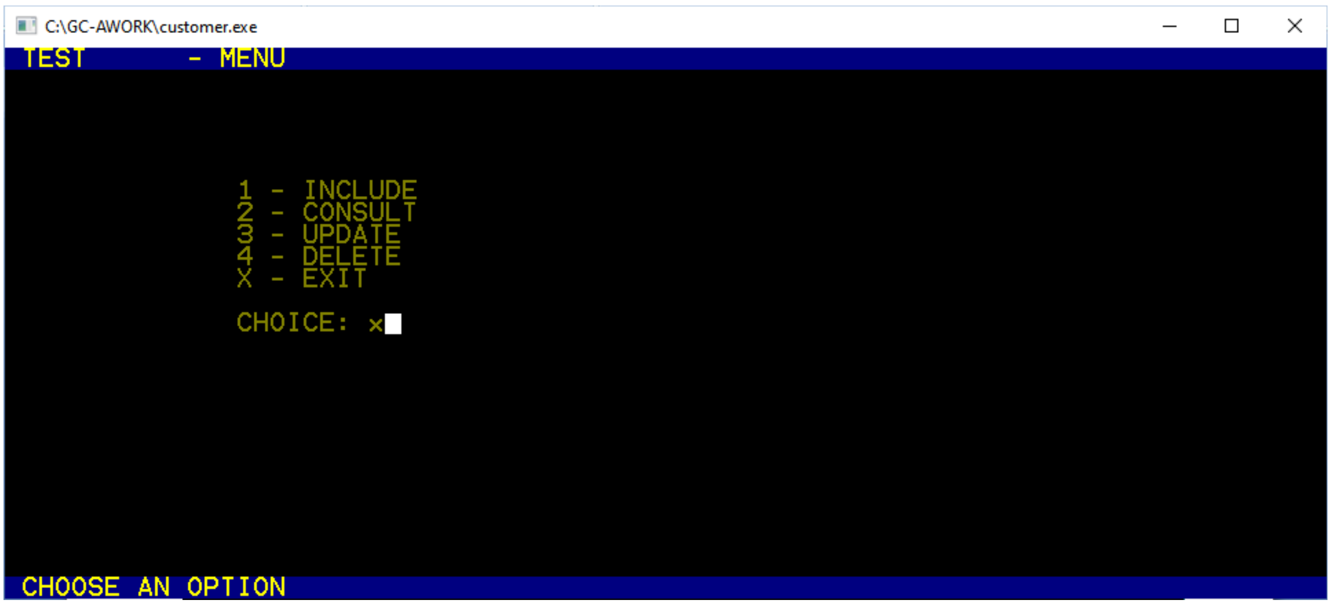
```
Prompt dei comandi - q3
CobGDB                  GnuCOBOL  GDB  Interpreter                ▓ ▶ → ↓ ↑ ■ ↔ D ?
  100                   10 COLUMN PLUS 2 TO WS-ERROR.
  101
  102           PROCEDURE DIVISION.
B 103           001-START.
  104               SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
  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
B 116                   DISPLAY SS-CLS
 !117                   ACCEPT SS-MENU
  118
  119                   EVALUATE TRUE
  120                       WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
page up
```

2.9. Focus Command

The application is running in a separate window.
The 'O = Focus' command switches the focus from debugging / animation window to the application window.
Instead of using the "O" command you can left-click on the <-> icon.

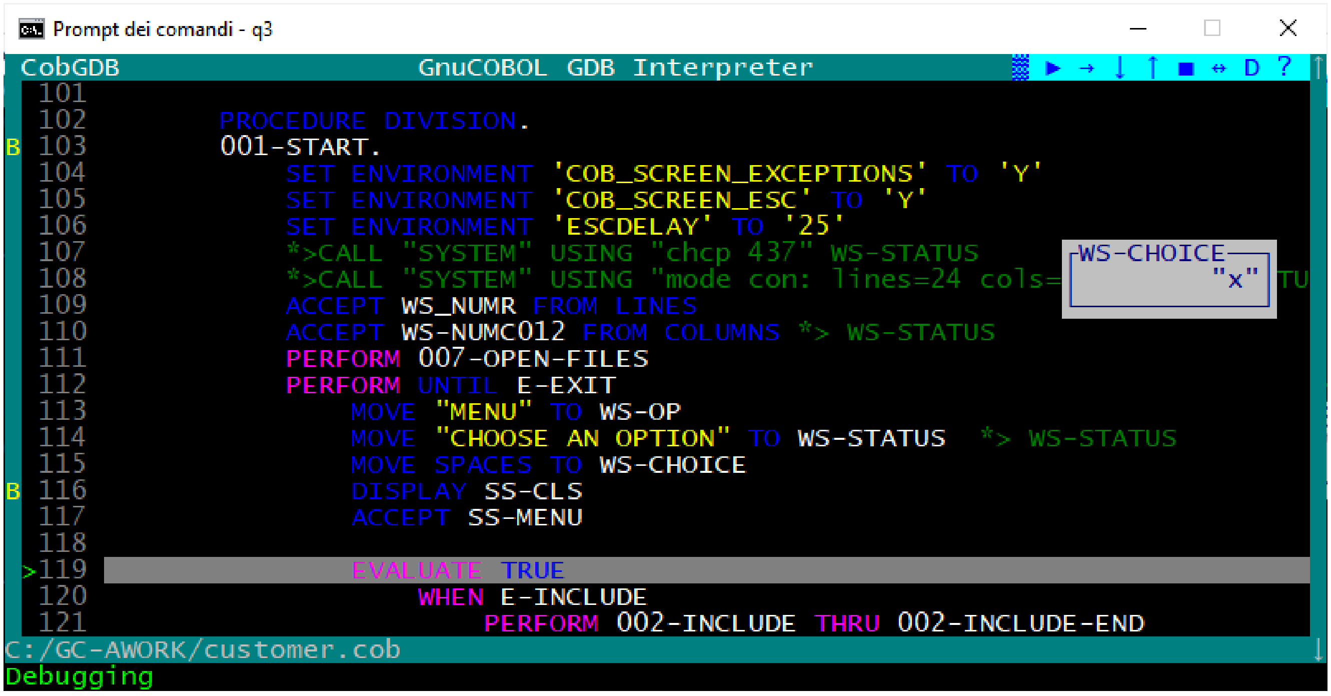

```
Prompt dei comandi - q3
CobGDB                  GnuCOBOL GDB Interpreter
               10 COLUMN PLUS 2 TO WS-ERROR.

       PROCEDURE DIVISION.
       001-START.
           SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
           SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
           SET ENVIRONMENT 'ESCDELAY' TO '25'
           *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
           *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
           ACCEPT WS_NUMR FROM LINES
           ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
           PERFORM 007-OPEN-FILES
           PERFORM UNTIL E-EXIT
               MOVE "MENU" TO WS-OP
               MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
               MOVE SPACES TO WS-CHOICE
               DISPLAY SS-CLS
               ACCEPT SS-MENU

               EVALUATE TRUE
                   WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
switch to the debug output
```

The focus now is on the application window where a user action is needed.
In our sample, we type the "X" choice and Enter:


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




              1 - INCLUDE
              2 - CONSULT
              3 - UPDATE
              4 - DELETE
              X - EXIT

              CHOICE:  x











 CHOOSE AN OPTION
```

DOCUMENT CODE	MODULE: xxxxxxxxxx	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	25	

Now the system automatic switch back to the debugger / animation window to continue debugging.
You can see that the ACCEPT statement has been executed, now you are on line >119 :

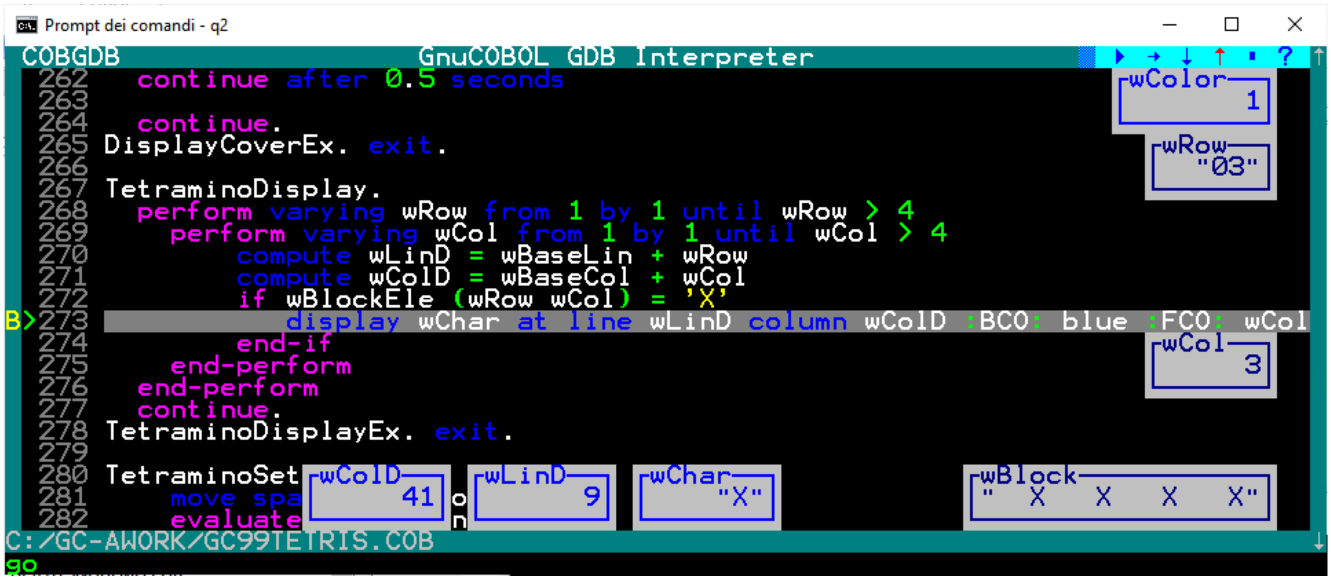


```
Prompt dei comandi - q3
 CobGDB                        GnuCOBOL  GDB Interpreter
  101
  102           PROCEDURE DIVISION.
B 103           001-START.
  104               SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
  105               SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
  106               SET ENVIRONMENT 'ESCDELAY' TO '25'
  107               *>CALL "SYSTEM" USING "chcp 437" WS-STATUS      ┌WS-CHOICE──┐
  108               *>CALL "SYSTEM" USING "mode con: lines=24 cols=│       "x"│TU
  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
  121                           PERFORM 002-INCLUDE THRU 002-INCLUDE-END
C:/GC-AWORK/customer.cob
Debugging
```

2.10.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:



```
Prompt dei comandi - q2
COBGDB               GnuCOBOL GDB Interpreter
    continue after 0.5 seconds                                   ┌wColor──┐
                                                                 │       1│
    continue.                                                     ┌wRow──┐
 DisplayCoverEx. exit.                                            │  "03"│

 TetraminoDisplay.
    perform varying wRow from 1 by 1 until wRow > 4
       perform varying wCol from 1 by 1 until wCol > 4
             compute wLinD = wBaseLin + wRow
             compute wColD = wBaseCol + wCol
             if wBlockEle (wRow wCol) = 'X'
B>              display wChar at line wLinD column wColD :BCO: blue :FCO: wCol
             end-if                                               ┌wCol──┐
       end-perform                                                │     3│
    end-perform
    continue.
 TetraminoDisplayEx. exit.

 TetraminoSet┌wColD───┐o┌wLinD───┐ ┌wChar───┐           ┌wBlock──────────┐
       move spa│    41│ │       9│ │     "X"│           │"  X   X   X   X"│
       evaluate        n
C:/GC-AWORK/GC99TETRIS.COB
go
```

2.11.File Command

To show this command we use following sample:

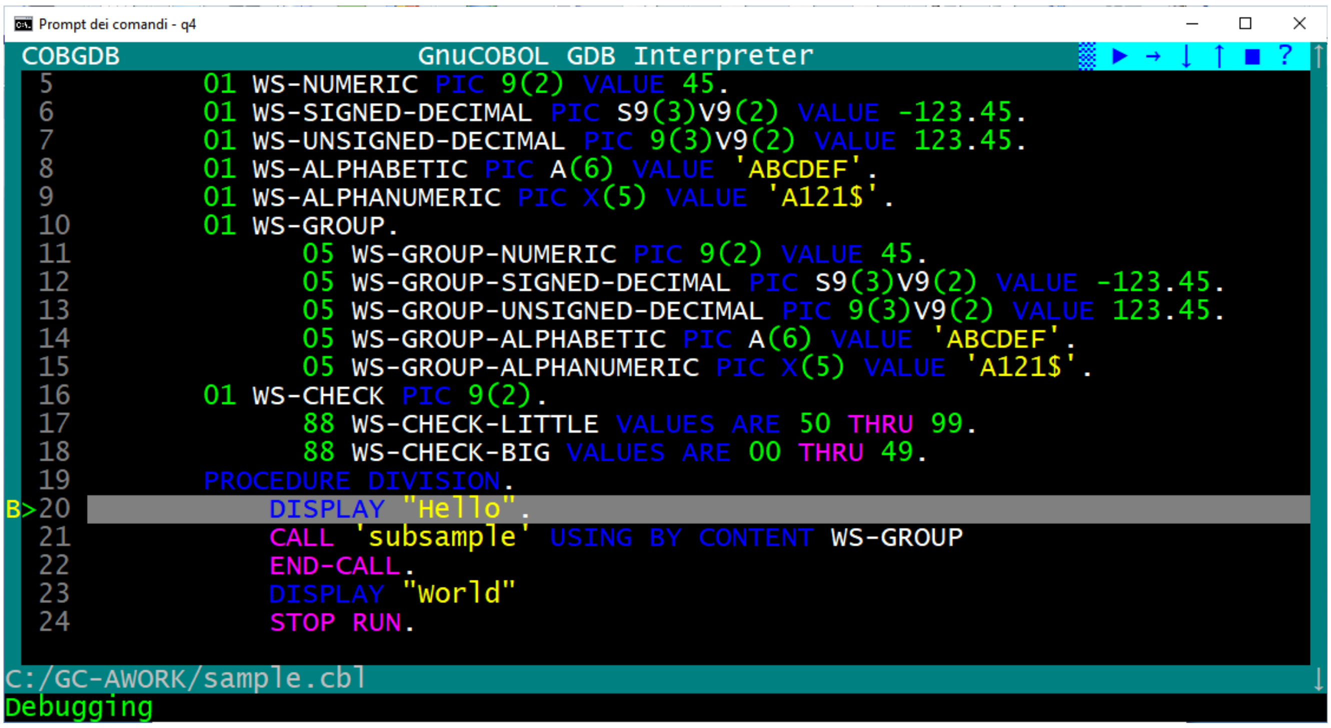
```
cobgdb sample.cbl subsample.cbl subsubsample.cbl -x -lpdcurses
```

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.

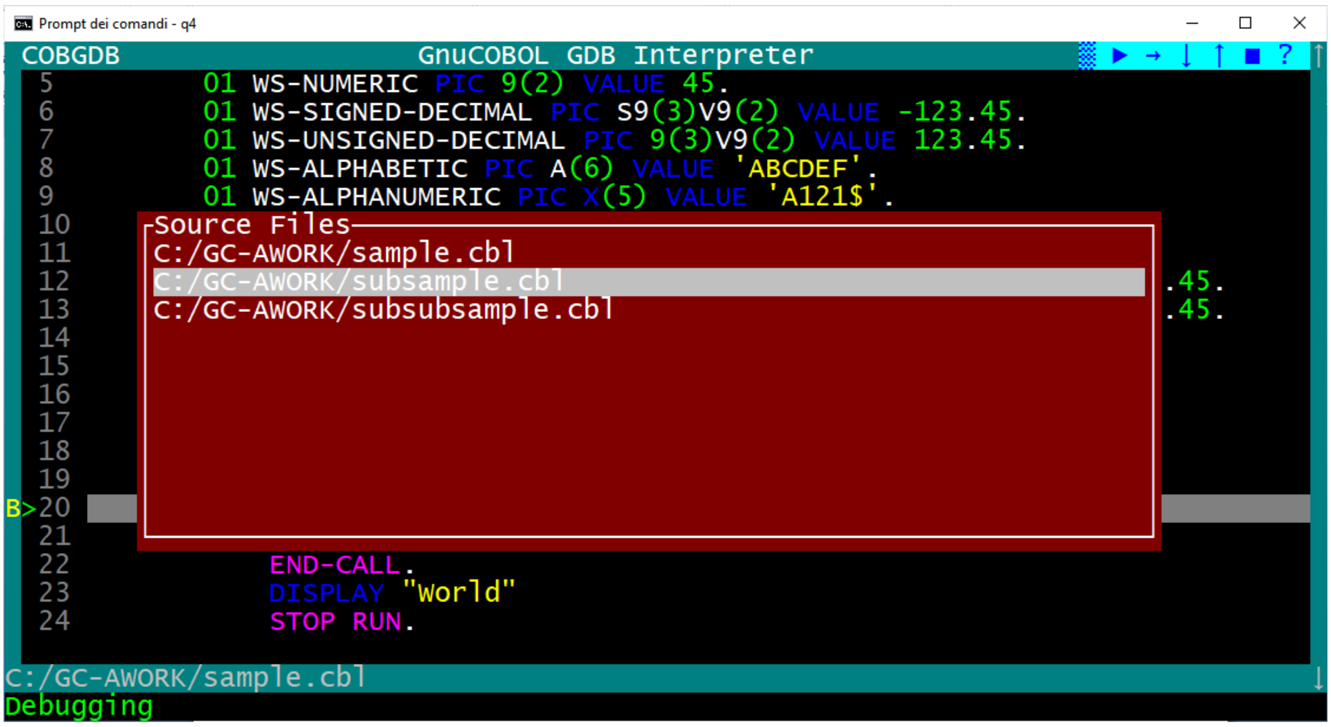
This example shows that when you need to debug only subsample.cbl or only subsubsample.cbl you need to execute COBGDB with all three programs.

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.



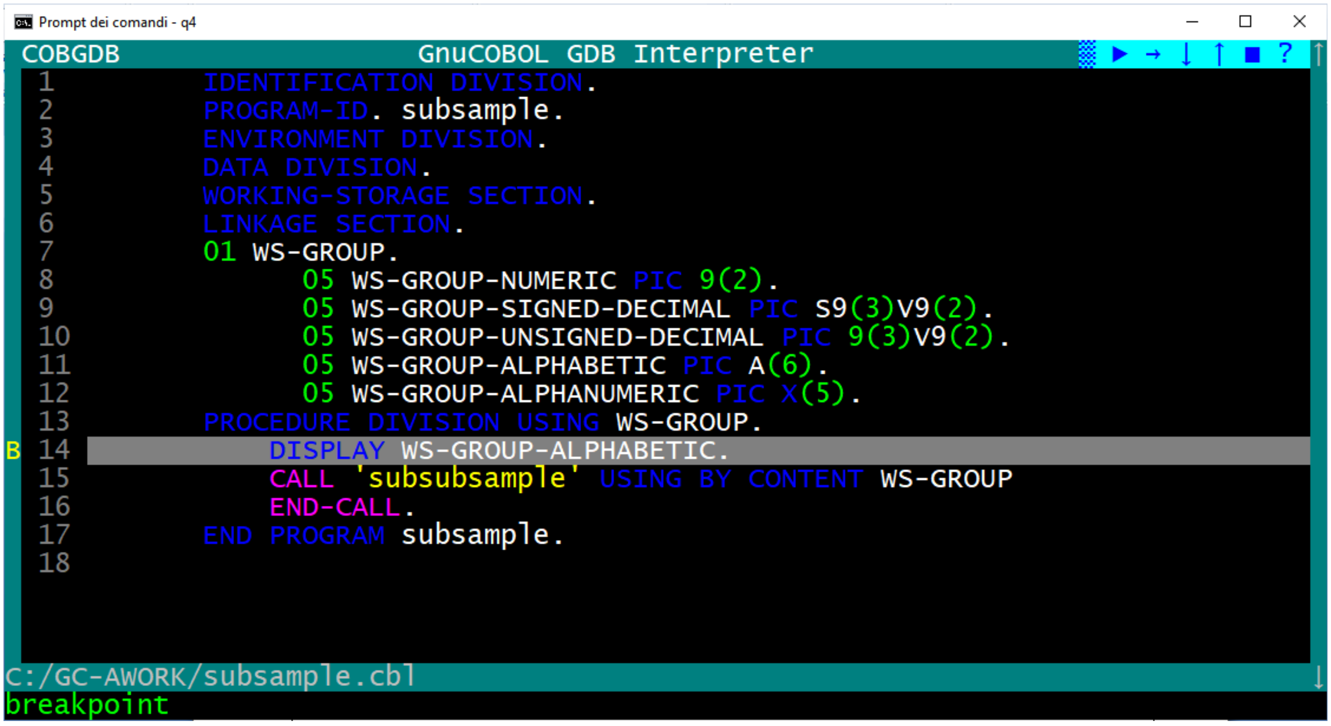
```
Prompt dei comandi - q4
COBGDB                   GnuCOBOL GDB Interpreter
    01 WS-NUMERIC PIC 9(2) VALUE 45.
    01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
    01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
    01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
    01 WS-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
    01 WS-GROUP.
        05 WS-GROUP-NUMERIC PIC 9(2) VALUE 45.
        05 WS-GROUP-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
        05 WS-GROUP-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
        05 WS-GROUP-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
        05 WS-GROUP-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
    01 WS-CHECK PIC 9(2).
        88 WS-CHECK-LITTLE VALUES ARE 50 THRU 99.
        88 WS-CHECK-BIG VALUES ARE 00 THRU 49.
    PROCEDURE DIVISION.
B>      DISPLAY "Hello".
        CALL 'subsample' USING BY CONTENT WS-GROUP
        END-CALL.
        DISPLAY "World"
        STOP RUN.

C:/GC-AWORK/sample.cbl
Debugging
```

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.



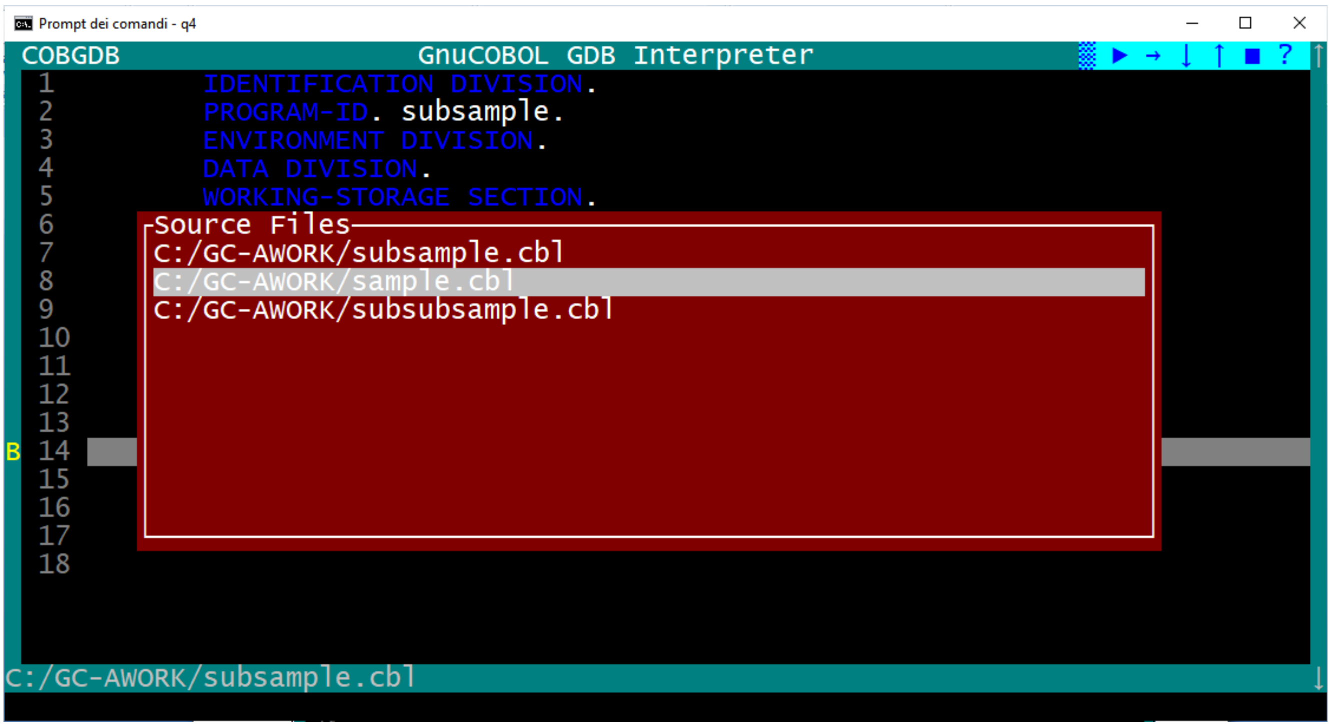
```
COBGDB                  GnuCOBOL GDB Interpreter
  5         01 WS-NUMERIC PIC 9(2) VALUE 45.
  6         01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
  7         01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
  8         01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
  9         01 WS-ALPHANUMERIC 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.
 ...
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.

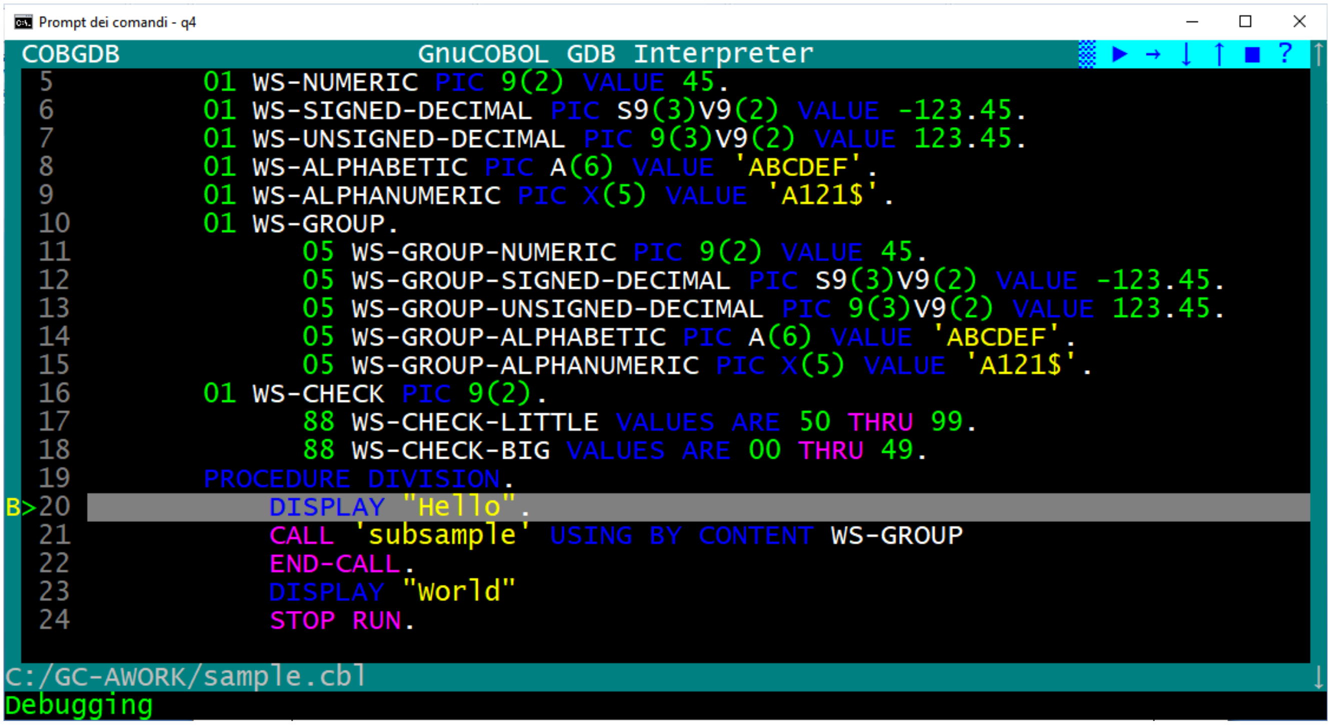


```
COBGDB                  GnuCOBOL GDB Interpreter
  1         IDENTIFICATION DIVISION.
  2         PROGRAM-ID. subsample.
  3         ENVIRONMENT DIVISION.
  4         DATA DIVISION.
  5         WORKING-STORAGE SECTION.
  6         LINKAGE SECTION.
  7         01 WS-GROUP.
  8               05 WS-GROUP-NUMERIC 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

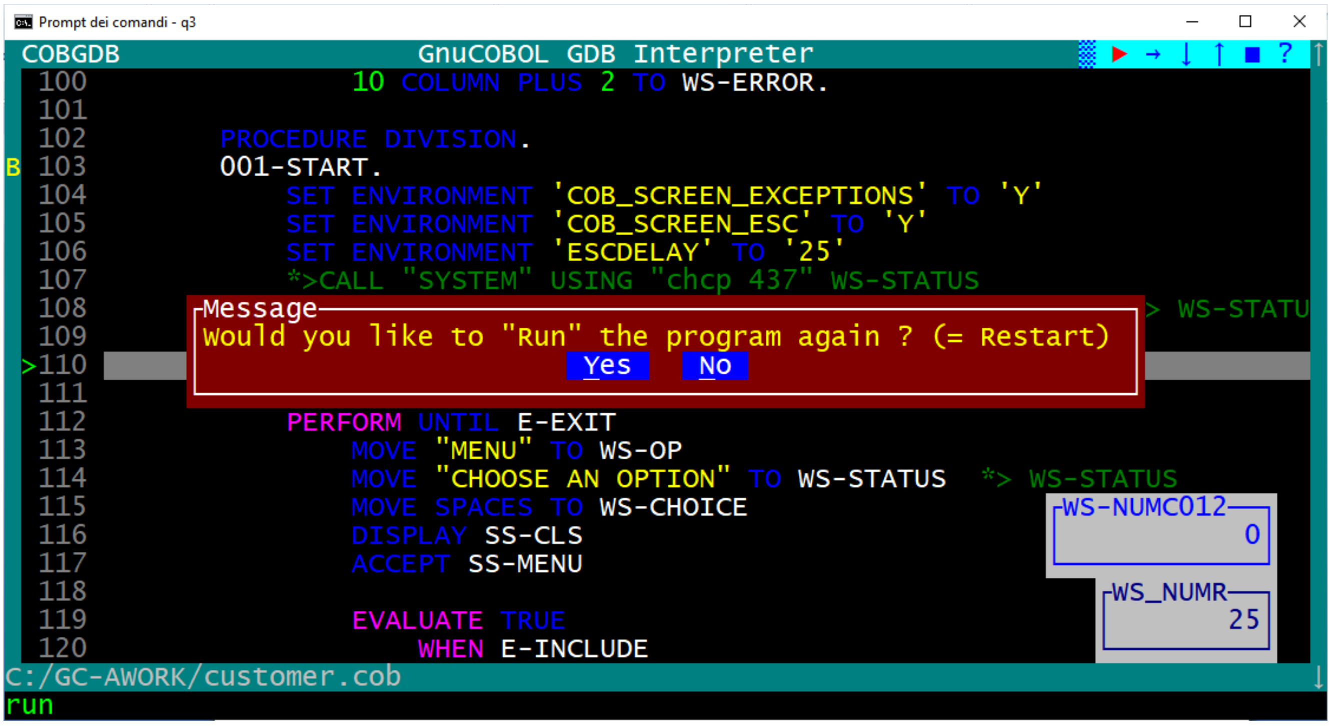

```
Prompt dei comandi - q4
COBGDB                    GnuCOBOL GDB Interpreter
           IDENTIFICATION DIVISION.
           PROGRAM-ID. subsample.
           ENVIRONMENT DIVISION.
           DATA DIVISION.
           WORKING-STORAGE SECTION.
        ┌Source Files─────────────────────────────────────────────┐
        │C:/GC-AWORK/subsample.cbl                                │
        │C:/GC-AWORK/sample.cbl                                   │
        │C:/GC-AWORK/subsubsample.cbl                             │
B       │                                                         │
        └─────────────────────────────────────────────────────────┘
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
           01 WS-NUMERIC PIC 9(2) VALUE 45.
           01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
           01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
           01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
           01 WS-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
           01 WS-GROUP.
                 05 WS-GROUP-NUMERIC PIC 9(2) VALUE 45.
                 05 WS-GROUP-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
                 05 WS-GROUP-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
                 05 WS-GROUP-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
                 05 WS-GROUP-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
           01 WS-CHECK PIC 9(2).
                 88 WS-CHECK-LITTLE VALUES ARE 50 THRU 99.
                 88 WS-CHECK-BIG VALUES ARE 00 THRU 49.
           PROCEDURE DIVISION.
B>             DISPLAY "Hello".
               CALL 'subsample' USING BY CONTENT WS-GROUP
               END-CALL.
               DISPLAY "World"
               STOP RUN.
C:/GC-AWORK/sample.cbl
Debugging
```

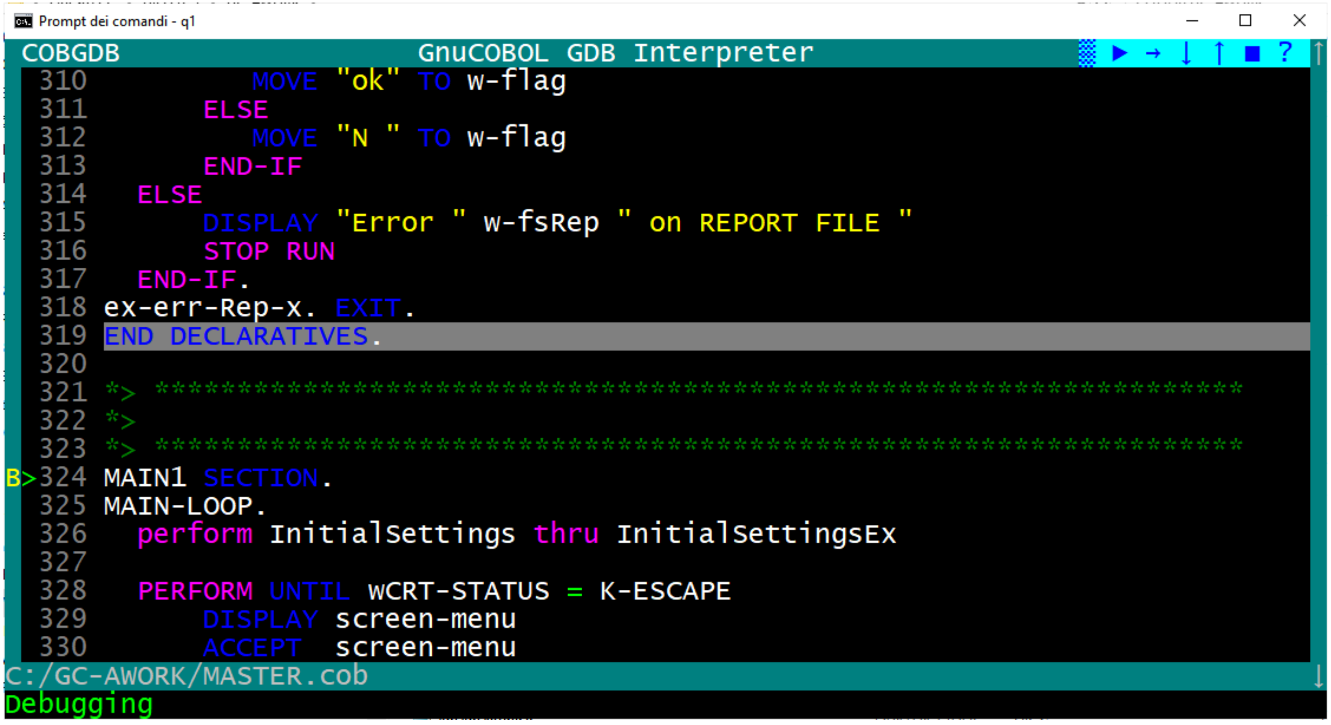
2.12.Run Command

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
               10 COLUMN PLUS 2 TO WS-ERROR.

       PROCEDURE DIVISION.
B      001-START.
           SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
           SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
           SET ENVIRONMENT 'ESCDELAY' TO '25'
           *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
 ┌Message                                                > WS-STATU
 │Would you like to "Run" the program again ? (= Restart)
>│                     Yes    No

           PERFORM UNTIL E-EXIT
               MOVE "MENU" TO WS-OP
               MOVE "CHOOSE AN OPTION" TO WS-STATUS   *> WS-STATUS
               MOVE SPACES TO WS-CHOICE                ┌WS-NUMC012─┐
               DISPLAY SS-CLS                          │          0│
               ACCEPT SS-MENU                          └───────────┘
                                                          ┌WS_NUMR─┐
               EVALUATE TRUE                              │      25│
                   WHEN E-INCLUDE                         └────────┘
C:/GC-AWORK/customer.cob
run
```

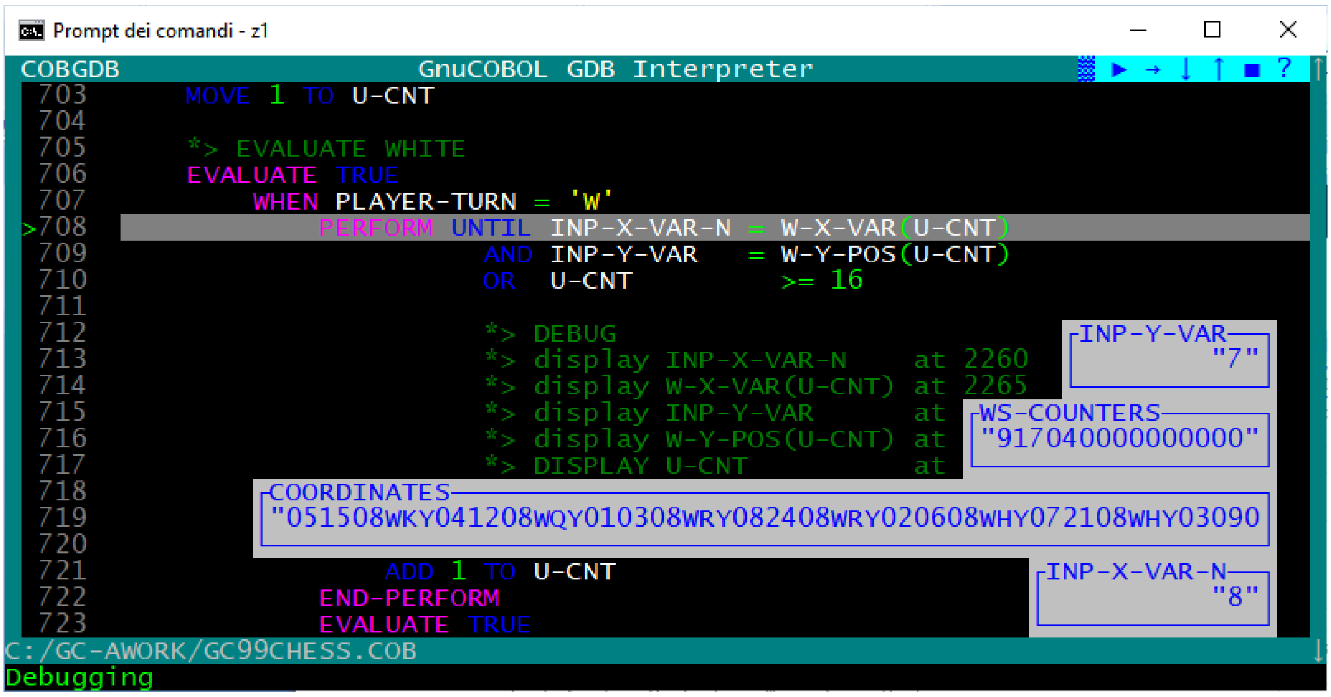
Note: if program has DECLARATIVES then the first automatic B Breakpoint will be settled at first executable PROCEDURE DIVISION statement that is the one after END DECLARATIVES, see following sample:



```
Prompt dei comandi - q1
COBGDB                  GnuCOBOL GDB Interpreter
         MOVE "ok" TO w-flag
      ELSE
         MOVE "N " TO w-flag
      END-IF
  ELSE
      DISPLAY "Error " w-fsRep " on REPORT FILE "
      STOP RUN
  END-IF.
ex-err-Rep-x. EXIT.
END DECLARATIVES.

*> *****************************************************************
*>
*> *****************************************************************
B>MAIN1 SECTION.
MAIN-LOOP.
  perform InitialSettings thru InitialSettingsEx

  PERFORM UNTIL WCRT-STATUS = K-ESCAPE
      DISPLAY screen-menu
      ACCEPT  screen-menu
C:/GC-AWORK/MASTER.cob
Debugging
```

2.13.Window Size command

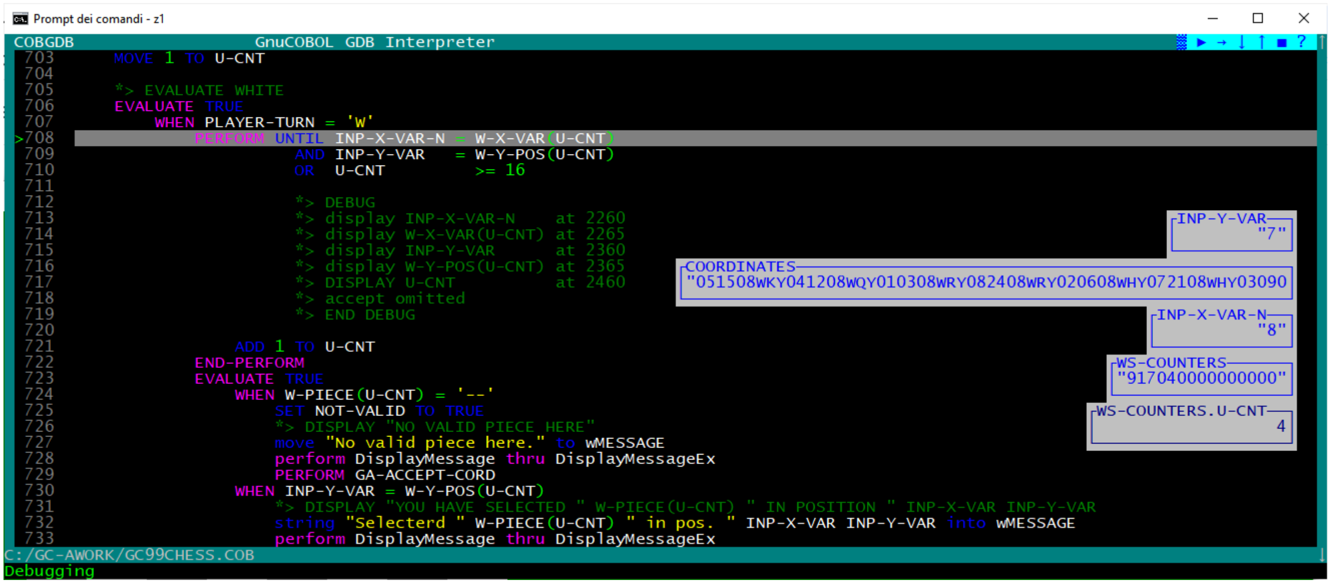
When you start a debugging session, the screen size is 24 x 80 columns.



```
Prompt dei comandi - z1
COBGDB                    GnuCOBOL  GDB  Interpreter
          MOVE  1  TO  U-CNT

          *>  EVALUATE  WHITE
          EVALUATE  TRUE
              WHEN  PLAYER-TURN  =  'W'
                  PERFORM  UNTIL  INP-X-VAR-N  =  W-X-VAR(U-CNT)
                                  AND  INP-Y-VAR    =  W-Y-POS(U-CNT)
                                  OR   U-CNT            >=  16

                                  *>  DEBUG
                                  *>  display  INP-X-VAR-N     at  2260
                                  *>  display  W-X-VAR(U-CNT)  at  2265
                                  *>  display  INP-Y-VAR       at
                                  *>  display  W-Y-POS(U-CNT)  at
                                  *>  DISPLAY  U-CNT           at

                          ADD  1  TO  U-CNT
                  END-PERFORM
                  EVALUATE  TRUE
C:/GC-AWORK/GC99CHESS.COB
Debugging
```

```
INP-Y-VAR
   "7"
WS-COUNTERS
"917040000000000"
COORDINATES
"051508WKY041208WQY010308WRY082408WRY020608WHY072108WHY03090
INP-X-VAR-N
   "8"
```

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

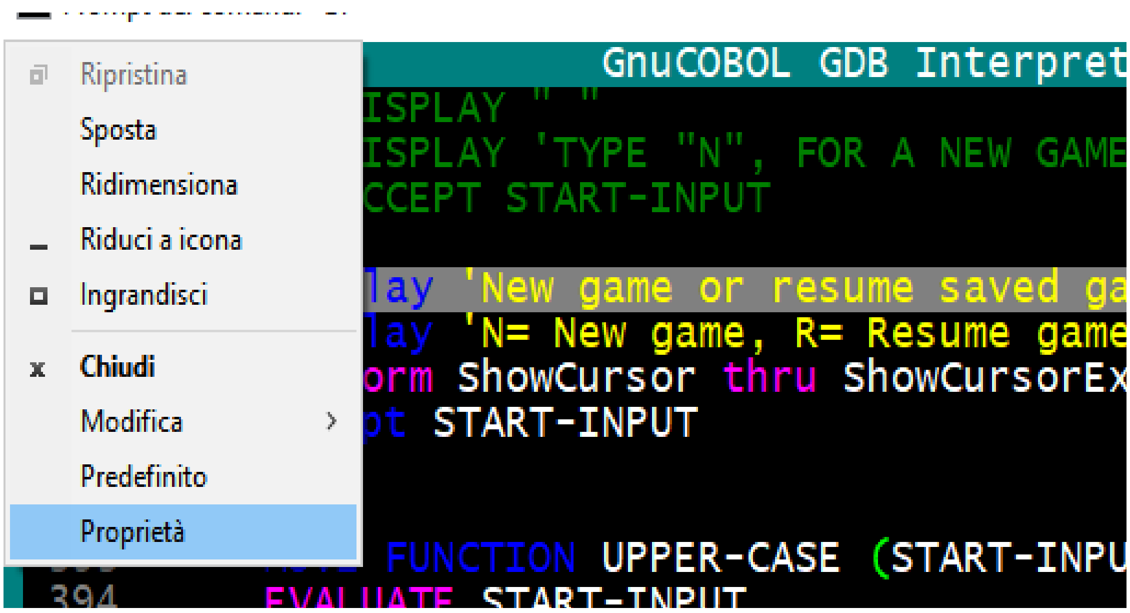


```
Prompt dei comandi - z1
COBGDB                    GnuCOBOL  GDB  Interpreter
          MOVE  1  TO  U-CNT

          *>  EVALUATE  WHITE
          EVALUATE  TRUE
              WHEN  PLAYER-TURN  =  'W'
                  PERFORM  UNTIL  INP-X-VAR-N  =  W-X-VAR(U-CNT)
                                  AND  INP-Y-VAR    =  W-Y-POS(U-CNT)
                                  OR   U-CNT            >=  16

                                  *>  DEBUG
                                  *>  display  INP-X-VAR-N     at  2260
                                  *>  display  W-X-VAR(U-CNT)  at  2265
                                  *>  display  INP-Y-VAR       at  2360
                                  *>  display  W-Y-POS(U-CNT)  at  2365
                                  *>  DISPLAY  U-CNT           at  2460
                                  *>  accept  omitted
                                  *>  END  DEBUG

                          ADD  1  TO  U-CNT
                  END-PERFORM
                  EVALUATE  TRUE
                      WHEN  W-PIECE(U-CNT)  =  '--'
                          SET  NOT-VALID  TO  TRUE
                          *>  DISPLAY  "NO  VALID  PIECE  HERE"
                          move  "No  valid  piece  here."  to  wMESSAGE
                          perform  DisplayMessage  thru  DisplayMessageEx
                          PERFORM  GA-ACCEPT-CORD
                      WHEN  INP-Y-VAR  =  W-Y-POS(U-CNT)
                          *>  DISPLAY  "YOU  HAVE  SELECTED  "  W-PIECE(U-CNT)  "  IN  POSITION  "  INP-X-VAR  INP-Y-VAR
                          string  "Selecterd  "  W-PIECE(U-CNT)  "  in  pos.  "  INP-X-VAR  INP-Y-VAR  into  wMESSAGE
                          perform  DisplayMessage  thru  DisplayMessageEx
C:/GC-AWORK/GC99CHESS.COB
Debugging
```

```
INP-Y-VAR
   "7"
COORDINATES
"051508WKY041208WQY010308WRY082408WRY020608WHY072108WHY03090
INP-X-VAR-N
   "8"
WS-COUNTERS
"917040000000000"
WS-COUNTERS.U-CNT
   4
```

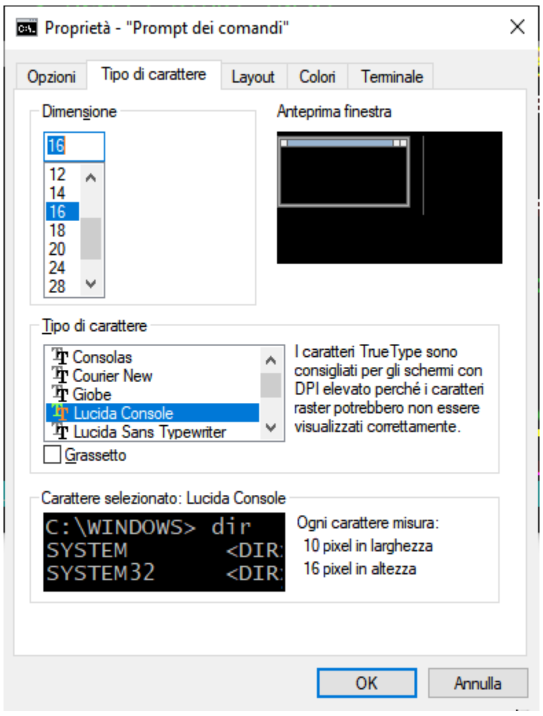
This can be very useful when you write GnuCOBOL code with the GnuCOBOL FREE FORMAT source option where each line of code can be longer than the classic 80 characters of GnuCOBOL FIXED FORMAT.

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



```
                GnuCOBOL GDB Interpret
ISPLAY " "
ISPLAY 'TYPE "N", FOR A NEW GAME
CCEPT START-INPUT

lay 'New game or resume saved ga
lay 'N= New game, R= Resume game
orm ShowCursor thru ShowCursorEx
pt START-INPUT

     FUNCTION UPPER-CASE (START-INPU
394       EVALUATE START-INPUT
```

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



Dimensione: 16 (12, 14, 16, 18, 20, 24, 28)

Tipo di carattere: Consolas, Courier New, Gigi, Lucida Console, Lucida Sans Typewriter

I caratteri TrueType sono consigliati per gli schermi con DPI elevato perché i caratteri raster potrebbero non essere visualizzati correttamente.

Grassetto

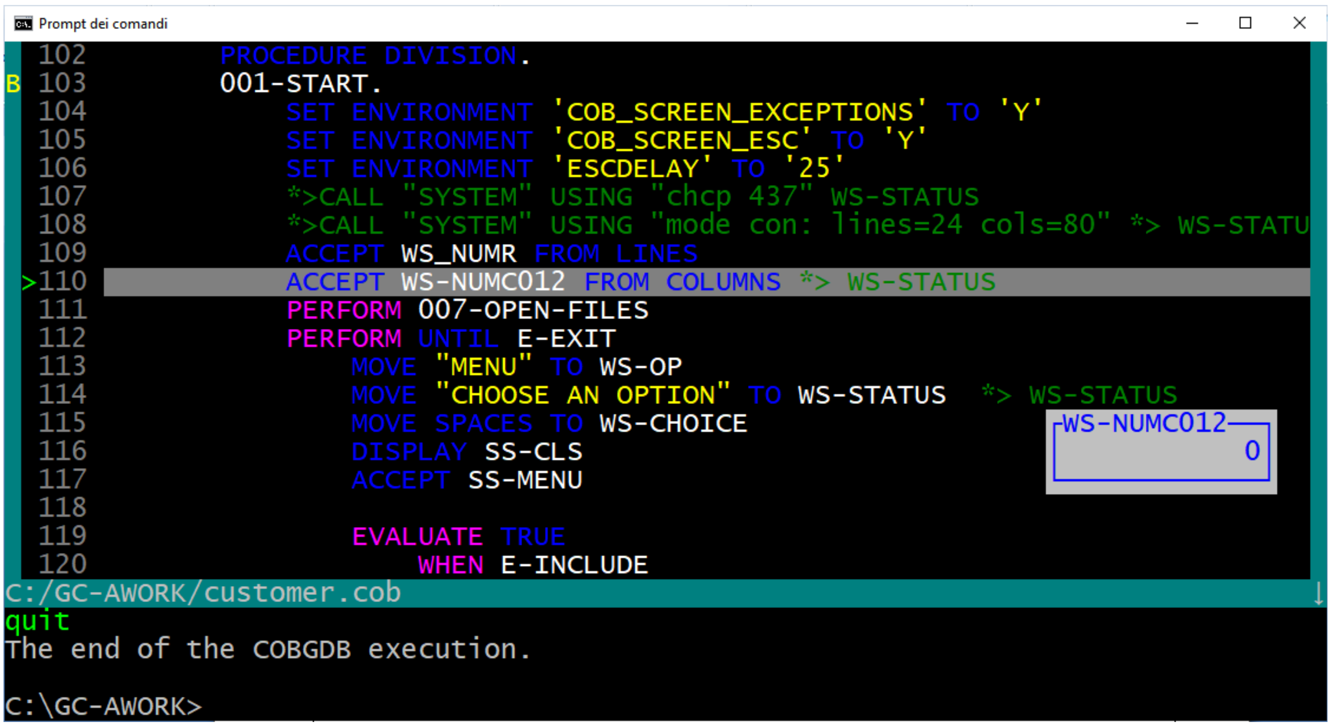
Carattere selezionato: Lucida Console

```
C:\WINDOWS> dir
SYSTEM          <DIR
SYSTEM32        <DIR
```

Ogni carattere misura:
10 pixel in larghezza
16 pixel in altezza

2.14.Quit Command

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



```
          PROCEDURE DIVISION.
B         001-START.
              SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
              SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
              SET ENVIRONMENT 'ESCDELAY' TO '25'
              *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
              *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATU
              ACCEPT WS_NUMR FROM LINES
>             ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
              PERFORM 007-OPEN-FILES
              PERFORM UNTIL E-EXIT
                  MOVE "MENU" TO WS-OP
                  MOVE "CHOOSE AN OPTION" TO WS-STATUS   *> WS-STATUS
                  MOVE SPACES TO WS-CHOICE                 ┌WS-NUMC012──┐
                  DISPLAY SS-CLS                           │           0│
                  ACCEPT SS-MENU                           └────────────┘

                  EVALUATE TRUE
                      WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
quit
The end of the COBGDB execution.

C:\GC-AWORK>
```

2.15.Attach Command

Prerequiste: your programs are all compiled for debug (use the -g, a cobc compiler option), because you just compile everything for debug because that's no big overhead and you can attach and use system corefiles at any time. COBGDB needs the .cob, .c, .c.h, and .c.l.h files in order to perform debugging.

Scenario 1. Start a program from the command line.
You are "somewhere in the application" and find the current state to be strange or can reproduce the bug by something you do. Now you just open the debugger from a different terminal, **Attach** to the program, debug normally, then either kill or detach.

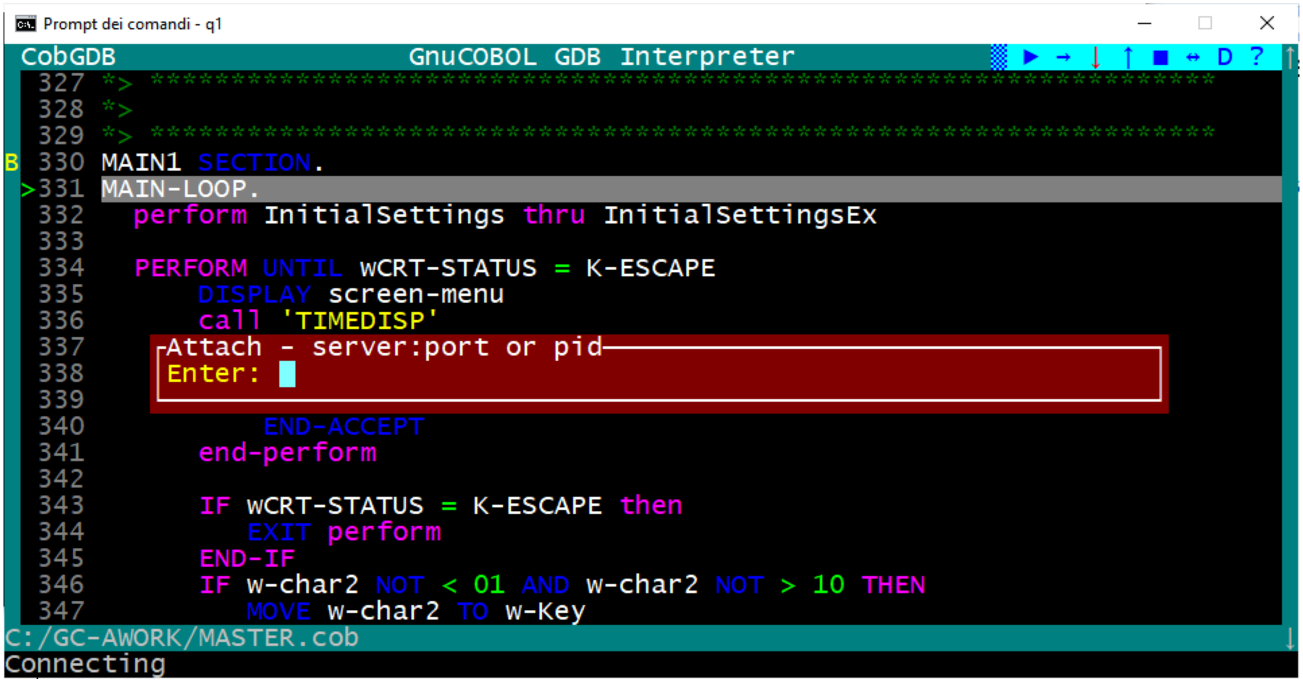
Scenario 2. The program has no terminal at all (like with [web]services, Java/C/HTML frontends, runs in the background, is started by other COBOL programs, ...) it is just "somehow started". Now you just open the debugger from a different terminal, **Attach** to the program, debug normally, then either kill or detach.

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

- `server:port = localhost:5555`
- `PID = 9112`

Command line:

- `cobgdb --connect localhost:5555 prog.cob`



```
Prompt dei comandi - q1
CobGDB                  GnuCOBOL GDB Interpreter
   *> ***********************************************************
   *>
   *> ***********************************************************
B  MAIN1 SECTION.
>  MAIN-LOOP.
     perform InitialSettings thru InitialSettingsEx

     PERFORM UNTIL wCRT-STATUS = K-ESCAPE
         DISPLAY screen-menu
         call 'TIMEDISP'
       ┌Attach - server:port or pid──────────────────────────────
       │Enter: 
       └──────────────────────────────────────────────────────────
             END-ACCEPT
         end-perform

         IF wCRT-STATUS = K-ESCAPE then
            EXIT perform
         END-IF
         IF w-char2 NOT < 01 AND w-char2 NOT > 10 THEN
            MOVE w-char2 TO w-Key
C:/GC-AWORK/MASTER.cob
Connecting
```

3. Other 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. Debugging sub programs

We use following sample programs `p0.cob` and `p1.cob` to show this feature:

```
>>SOURCE FORMAT IS FREE
IDENTIFICATION DIVISION.
PROGRAM-ID. p0.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  WS-NUMERIC                      PIC 9(2)       VALUE 45.
01  WS-SIGNED-DECIMAL               PIC S9(3)V9(2) VALUE -123.45.
01  WS-UNSIGNED-DECIMAL             PIC 9(3)V9(2)  VALUE 123.45.
01  WS-ALPHABETIC                   PIC A(6)       VALUE 'ABCDEF'.
01  WS-ALPHANUMERIC                 PIC X(5)       VALUE 'A121$'.
01  WS-GROUP.
    05  WS-GROUP-NUMERIC            PIC  9(2)      VALUE 45.
    05  WS-GROUP-SIGNED-DECIMAL     PIC S9(3)V9(2) VALUE -123.45.
    05  WS-GROUP-UNSIGNED-DECIMAL   PIC  9(3)V9(2) VALUE 123.45.
    05  WS-GROUP-ALPHABETIC         PIC  A(6)      VALUE 'ABCDEF'.
    05  WS-GROUP-ALPHANUMERIC       PIC  X(5)      VALUE 'A121$'.
    05  WS-MSG                      PIC  X(70).
01  WS-CHECK PIC 9(2).
    88 WS-CHECK-LITTLE VALUES ARE 50 THRU 99.
    88 WS-CHECK-BIG    VALUES ARE 00 THRU 49.

PROCEDURE DIVISION.
    DISPLAY "PROGRAM P0"
    DISPLAY WS-GROUP-NUMERIC
    DISPLAY WS-GROUP-SIGNED-DECIMAL
    DISPLAY WS-GROUP-UNSIGNED-DECIMAL
    DISPLAY WS-GROUP-ALPHABETIC
    DISPLAY WS-GROUP-ALPHANUMERIC
    DISPLAY WS-MSG

    MOVE 'Program "P1" was called by "P0"' TO WS-MSG
    CALL 'p1' USING BY CONTENT WS-GROUP END-CALL

    DISPLAY 'BACK TO "P0", received the message:'
    DISPLAY WS-GROUP-NUMERIC
    DISPLAY WS-GROUP-SIGNED-DECIMAL
    DISPLAY WS-GROUP-UNSIGNED-DECIMAL
    DISPLAY WS-GROUP-ALPHABETIC
    DISPLAY WS-GROUP-ALPHANUMERIC
    DISPLAY WS-MSG

    STOP RUN.
```

```
>>SOURCE FORMAT IS FREE
IDENTIFICATION DIVISION.
PROGRAM-ID. p1.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.
01 WS-GROUP.
    05 WS-GROUP-NUMERIC            PIC 9(2).
    05 WS-GROUP-SIGNED-DECIMAL     PIC S9(3)V9(2).
    05 WS-GROUP-UNSIGNED-DECIMAL   PIC 9(3)V9(2).
    05 WS-GROUP-ALPHABETIC         PIC A(6).
    05 WS-GROUP-ALPHANUMERIC       PIC X(5).
    05 WS-MSG                      PIC X(70).

PROCEDURE DIVISION USING WS-GROUP.
    DISPLAY "PROGRAM P1"
    DISPLAY "Message received: " WS-MSG
    DISPLAY 'BACK IN "P1" receiving the message:'
    DISPLAY WS-MSG
    GOBACK.
END PROGRAM p1.
```

DOCUMENT CODE	MODULE: xxxxxxxxxx	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	37	

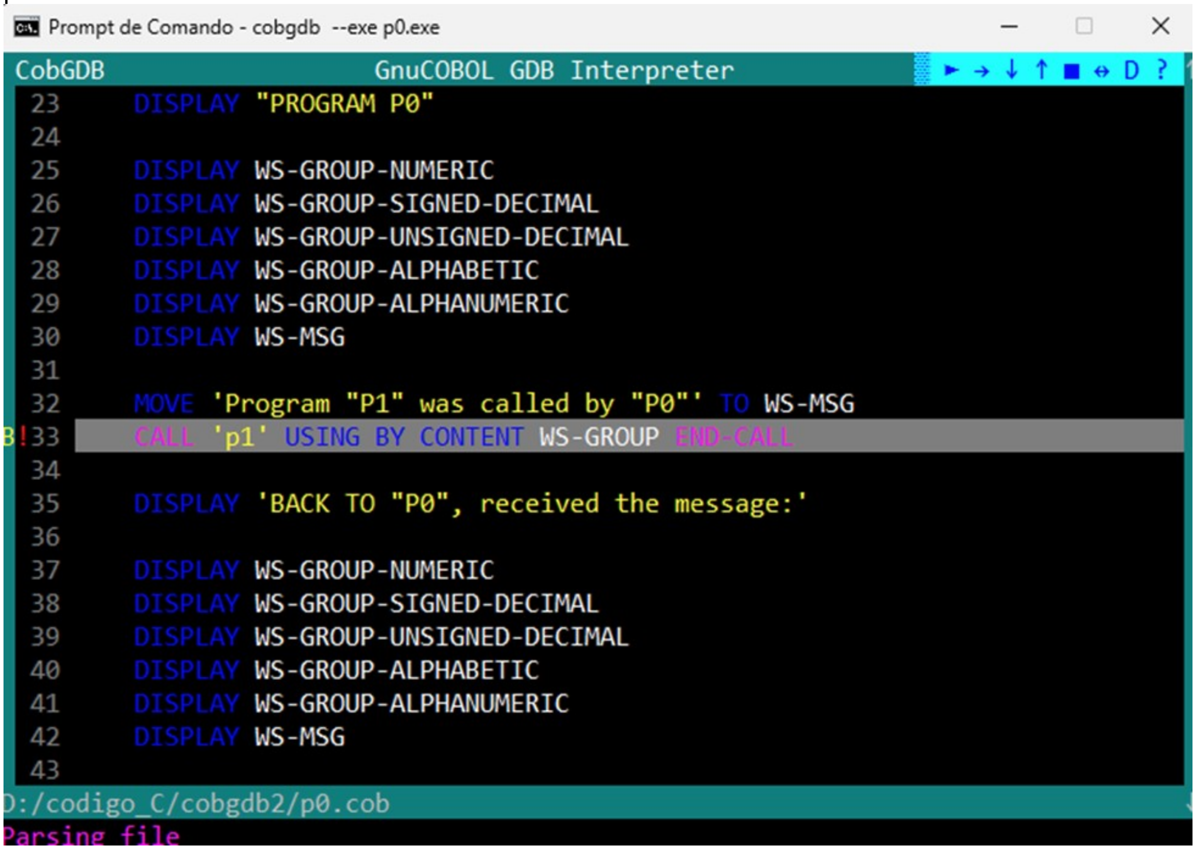
It is possible to debug modules as long as you compile them in the same directory as your source code, using the following command:

```
cobc -g -fsource-location -ftraceall -v -O0 -x p0.cob
cobc -g -fsource-location -ftraceall -v -O0 -m p1.cob
```

In this example, p0.cob is the source code of the executable, and p1.cob is the source code of the module (.dll on Windows or .so on Linux). Debug Execution:

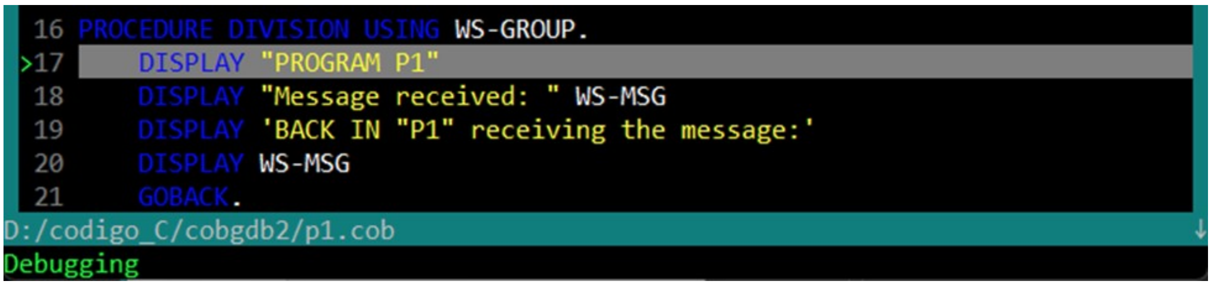
```
cobgdb --exe p0.exe
```

Example:



```
Prompt de Comando - cobgdb  --exe p0.exe
CobGDB                  GnuCOBOL GDB Interpreter
       DISPLAY "PROGRAM P0"

       DISPLAY WS-GROUP-NUMERIC
       DISPLAY WS-GROUP-SIGNED-DECIMAL
       DISPLAY WS-GROUP-UNSIGNED-DECIMAL
       DISPLAY WS-GROUP-ALPHABETIC
       DISPLAY WS-GROUP-ALPHANUMERIC
       DISPLAY WS-MSG

       MOVE 'Program "P1" was called by "P0"' TO WS-MSG
B!     CALL 'p1' USING BY CONTENT WS-GROUP END-CALL

       DISPLAY 'BACK TO "P0", received the message:'

       DISPLAY WS-GROUP-NUMERIC
       DISPLAY WS-GROUP-SIGNED-DECIMAL
       DISPLAY WS-GROUP-UNSIGNED-DECIMAL
       DISPLAY WS-GROUP-ALPHABETIC
       DISPLAY WS-GROUP-ALPHANUMERIC
       DISPLAY WS-MSG

D:/codigo_C/cobgdb2/p0.cob
Parsing file
```

Resulting in:



```
 PROCEDURE DIVISION USING WS-GROUP.
>    DISPLAY "PROGRAM P1"
     DISPLAY "Message received: " WS-MSG
     DISPLAY 'BACK IN "P1" receiving the message:'
     DISPLAY WS-MSG
     GOBACK.
D:/codigo_C/cobgdb2/p1.cob
Debugging
```

If p1.cob is already compiled as a module with:

```
cobc -g -fsource-location -ftraceall -v -O0 -m p1.cob
```

the following command also works:

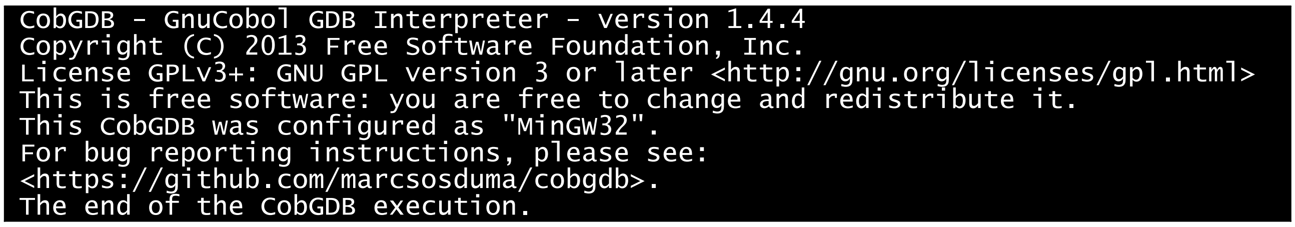
```
cobgdb p0.cob
```

When you are debugging separately compiled subprograms, as shown in the previous chapter, you can use the same **F File** command, which you should use when debugging subprograms that have all been compiled into the same executable. With **F File** command you can load the source code of a module and set a breakpoint.

3.3. COBGDB Version

Use command option: `cobgdb --version`

to display CobGDB version informations as follows:



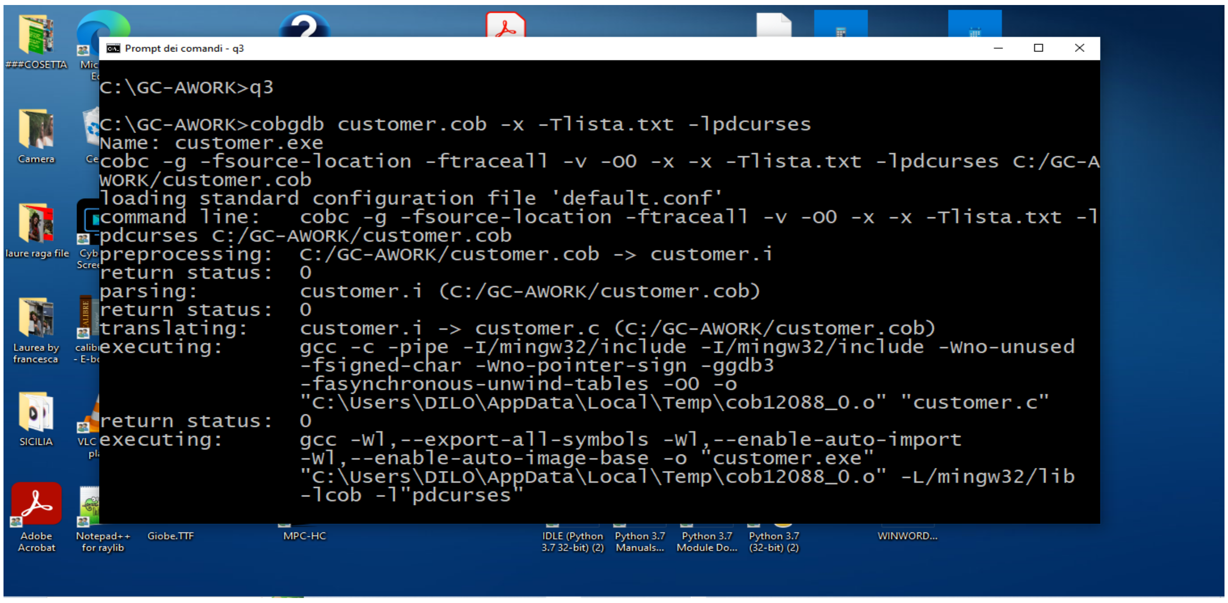
```
CobGDB - GnuCobol GDB Interpreter - version 1.4.4
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
This CobGDB was configured as "MinGW32".
For bug reporting 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 and icon useful to enable or disable automatic display variables during debugging / animation* . ***Version 8 of 2025.06.16.*** *Added the new O (Focus) command and icon useful to switch the focus to application screen from debugging / animation screen* . ***Version 9 of 2025.10.20.*** *Added more documentation for the A (Attach) command.* ***Version 10 of 2025.11.06.*** *Added "Debugging sub programs" Chapter.* *Added F = File command also for separately compiled subprograms* *Added CTRL-F and CTR-L commands*

DOCUMENT CODE	MODULE: XXXXXXXXXX	**USING COBGDB FOR GnuCOBOL**	PAGE	**GnuCOBOL**
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	40	

Technical info



```
C:\GC-AWORK>q3

C:\GC-AWORK>cobgdb customer.cob -x -Tlista.txt -lpdcurses
Name: customer.exe
cobc -g -fsource-location -ftraceall -v -O0 -x -x -Tlista.txt -lpdcurses C:/GC-A
WORK/customer.cob
loading standard configuration file 'default.conf'
command line:   cobc -g -fsource-location -ftraceall -v -O0 -x -x -Tlista.txt -l
pdcurses C:/GC-AWORK/customer.cob
preprocessing:  C:/GC-AWORK/customer.cob -> customer.i
return status:  0
parsing:        customer.i (C:/GC-AWORK/customer.cob)
return status:  0
translating:    customer.i -> customer.c (C:/GC-AWORK/customer.cob)
executing:      gcc -c -pipe -I/mingw32/include -I/mingw32/include -Wno-unused
                -fsigned-char -Wno-pointer-sign -ggdb3
                -fasynchronous-unwind-tables -O0 -o
                "C:\Users\DILO\AppData\Local\Temp\cob12088_0.o" "customer.c"
return status:  0
executing:      gcc -Wl,--export-all-symbols -Wl,--enable-auto-import
                -Wl,--enable-auto-image-base -o "customer.exe"
                "C:\Users\DILO\AppData\Local\Temp\cob12088_0.o" -L/mingw32/lib
                -lcob -l"pdcurses"
```