LISP Debugger Guide

GstarCAD 2027

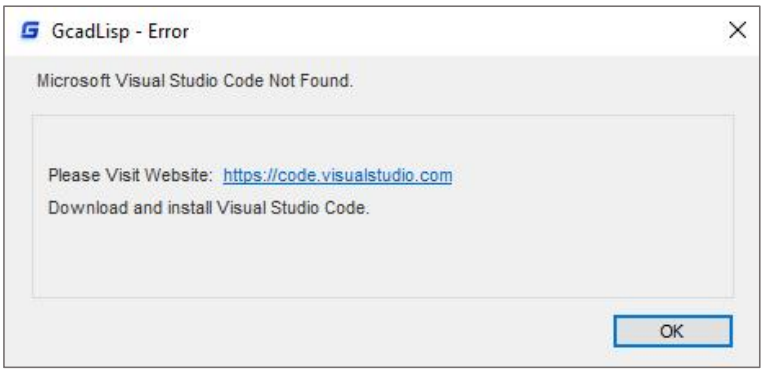
Table of Contents

1. Visual Studio Code

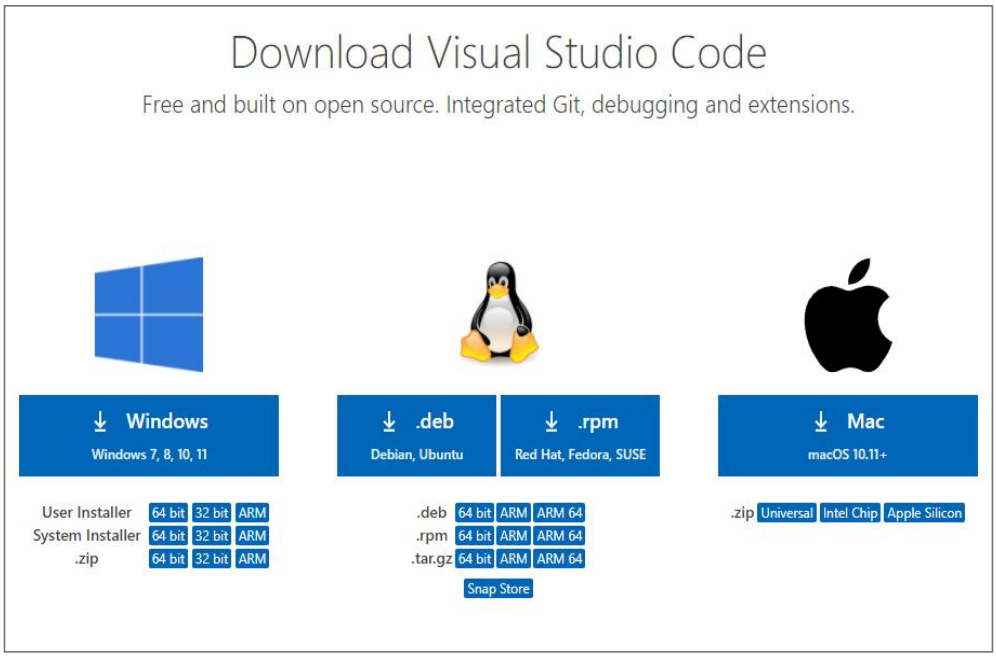
1.1. Install Visual Studio Code (VS Code)

Visual Studio Code (VS Code) is a free cross-platform source code editor developed by Microsoft®. The software supports functions such as syntax highlighting, intelligent code completion, code refactoring, and viewing definitions, and has built-in command-line tools and Git version control system.

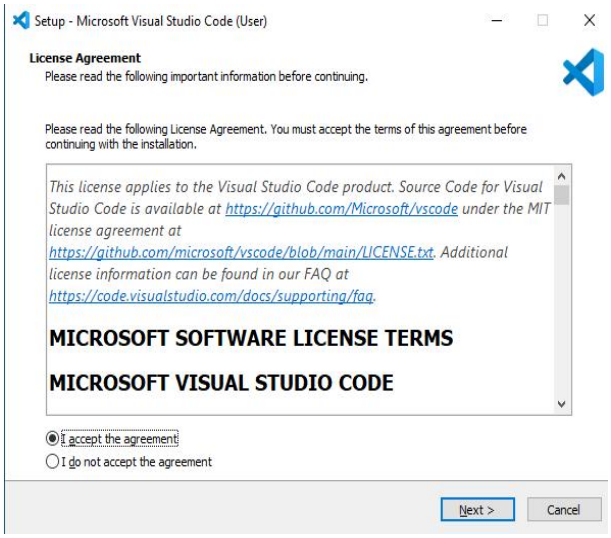
With the functions provided by VS Code, it is possible to debug the program code in GstarCAD. Enter '*VLIDE*' or '*VLISP*' at the command line of GstarCAD to call up VS Code. If VS Code is not installed, the following message appears:

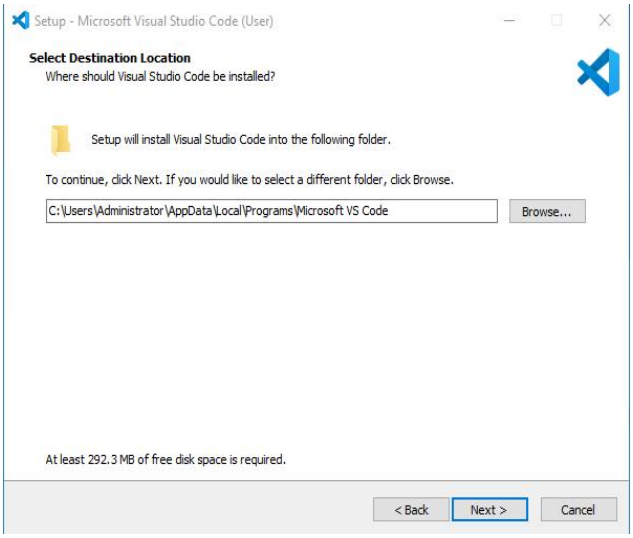


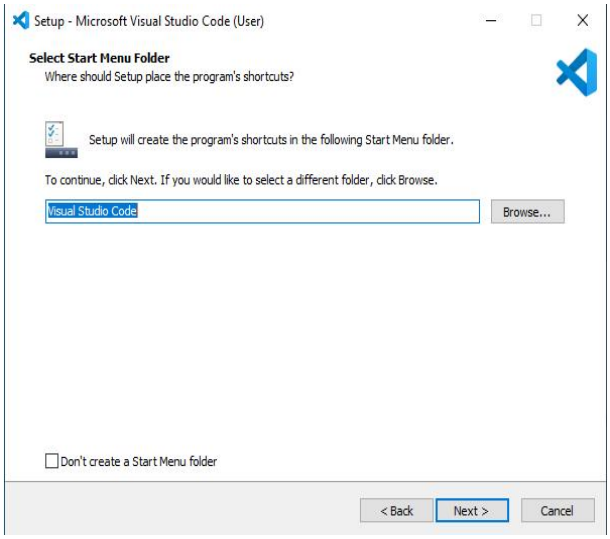
Click the link https://code.visualstudio.com/ in the above window or visit the official website of VS Code directly to download and install VS Code.

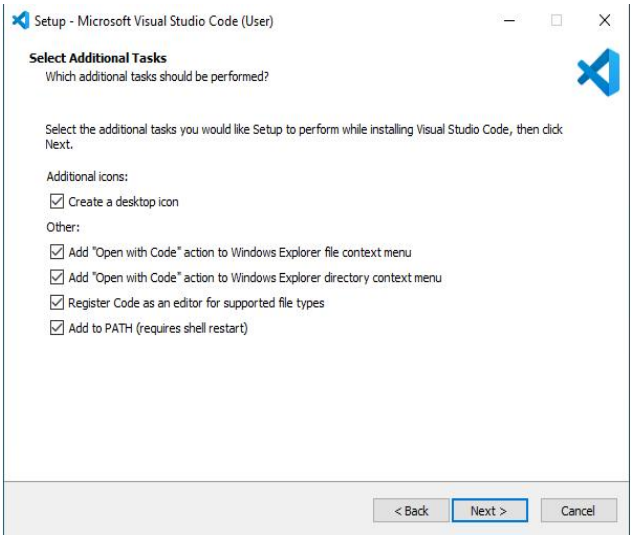


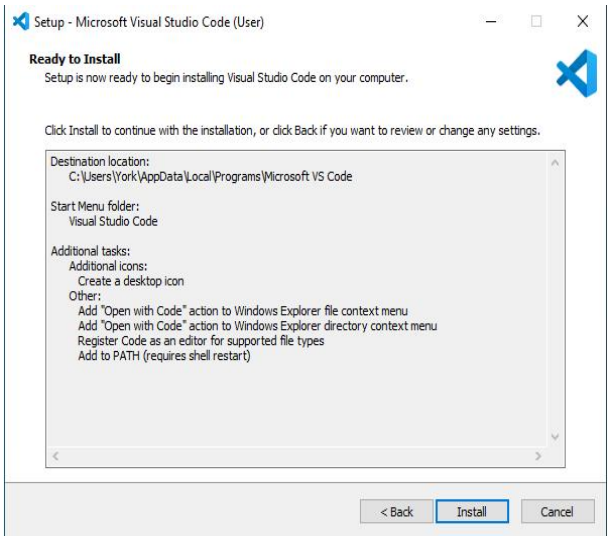
As an example, the following shows the process of installing VS Code for Windows after download.

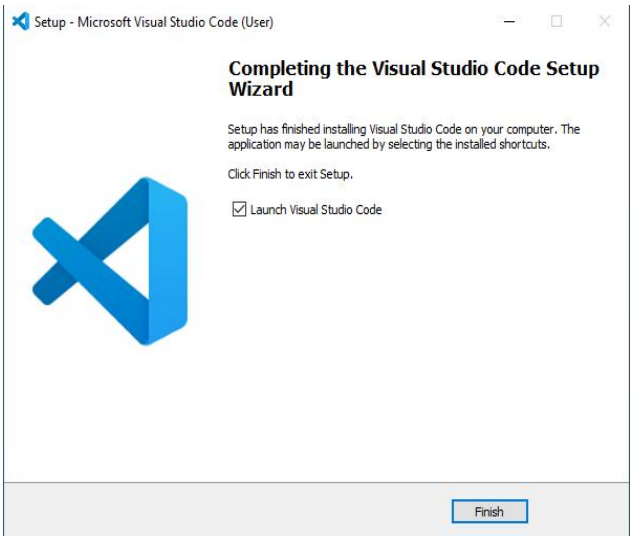










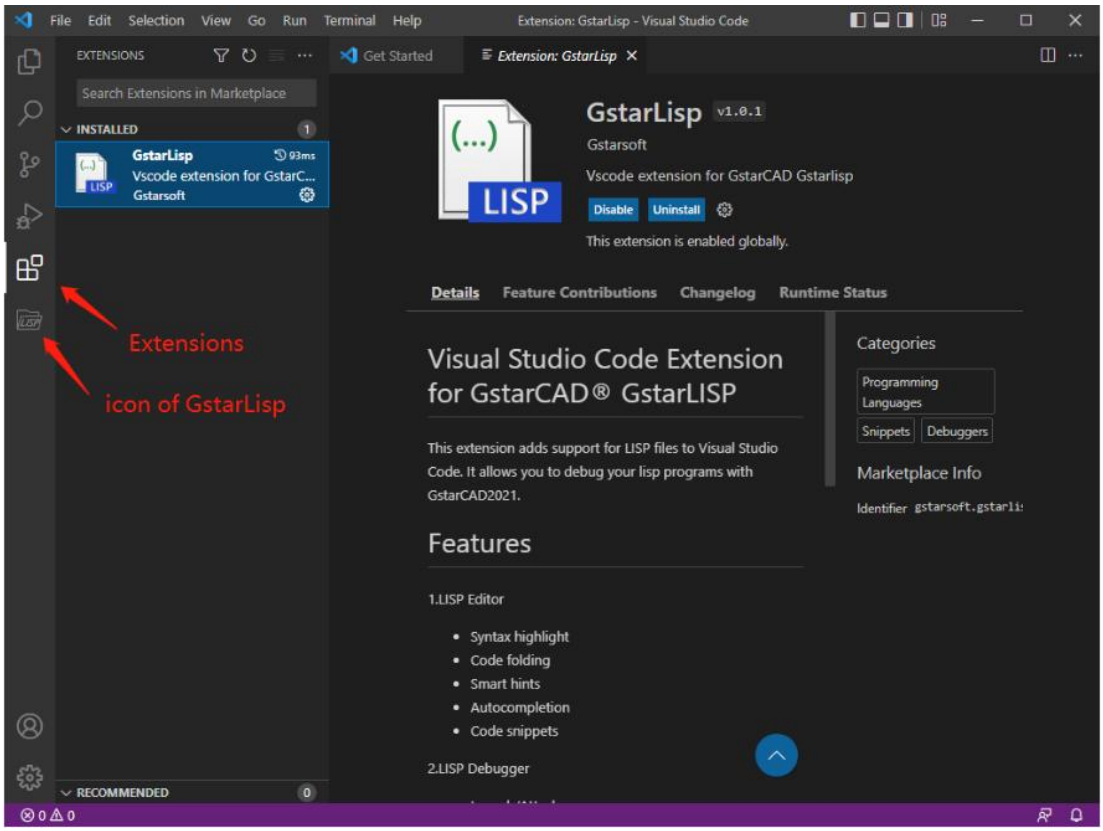


1.2. Install GstarLisp Plug-in

When starting VS Code through GstarCAD, **GstarLisp** plug-in is automatically installed by default. If there is a new version of **GstarLisp**, it will be automatically updated next time when starting VS Code through GstarCAD.

GstarLisp plug-in can also be installed manually. In this case search '*GstarCAD*', '*CAD*' or "*LISP*" in the VS Code application store, select '*GstarLisp*' and click to install.

After the plug-in is installed successfully, the icon of the **GstarLisp** is displayed in the activity bar. Details of **GstarLisp**, including name, introduction, features, new functions, installation, user guide, license information, change log, etc. can also be viewed as shown below.



1.3. Visual Studio Code (VS Code) User Interface

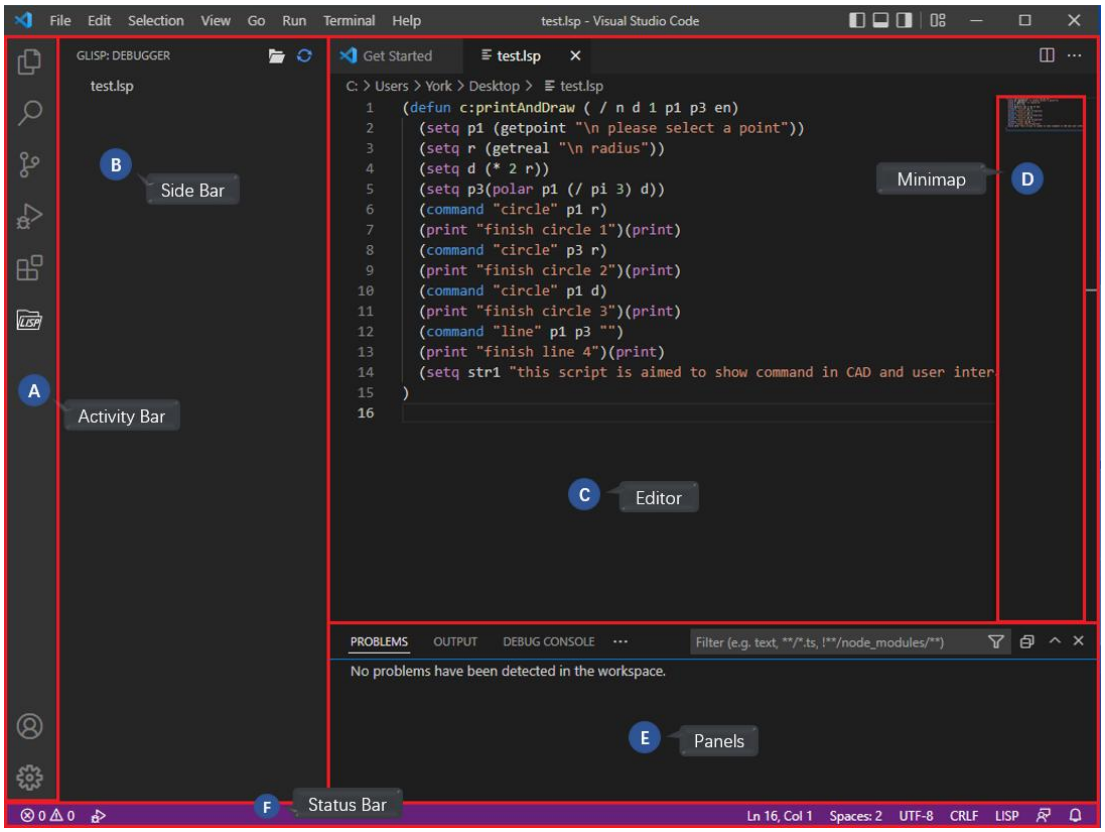
After VS Code and **GstarLisp** plug-in are installed, input '*VLIDE*' or '*VLISP*' at the GstarCAD command line to start VS Code.

```
VLISP
Command: Now installing GstarCAD Visual LISP Editor, It will take a few time.
GstarCAD Visual LISP Editor installed, And launching Visual Studio Code.
```

The following screenshot shows the user interface of VS Code. To learn more details about the user interface please visit the official

website of VS Code at:

https://code.visualstudio.com/docs/getstarted/userinterface



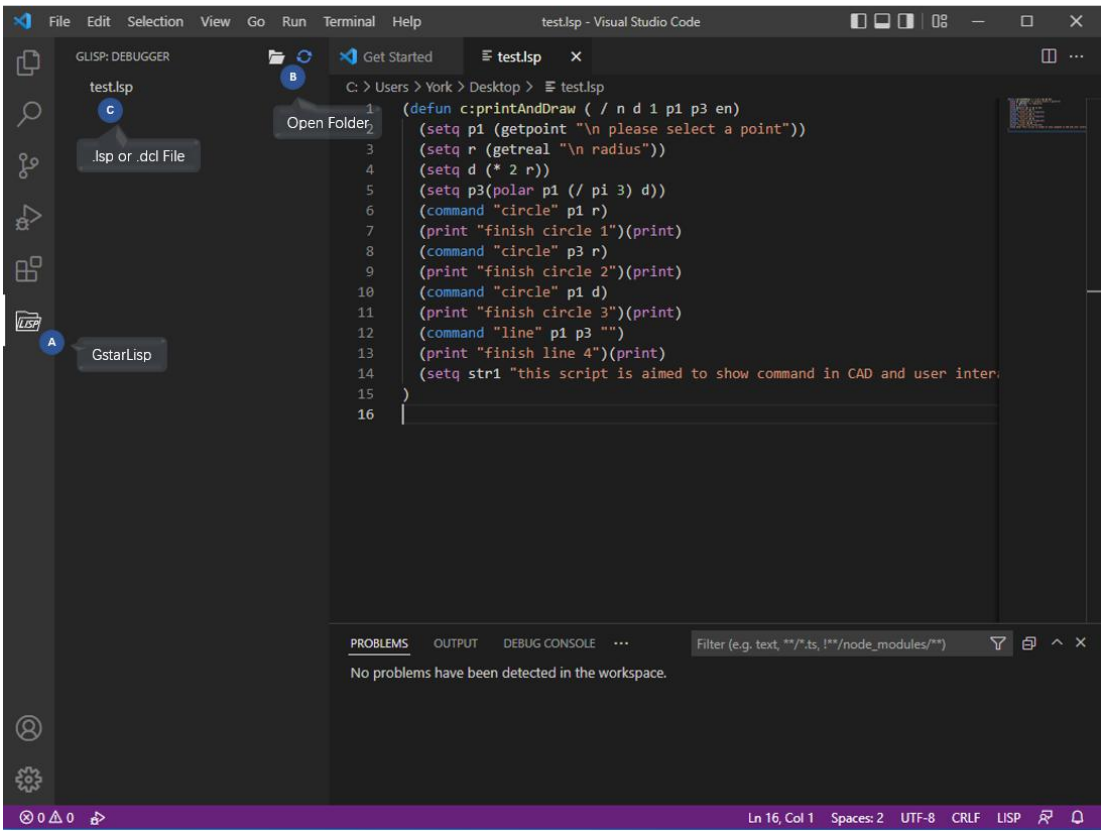
- **Activity Bar (A)**: Located on the far left side. Allows switching between views, debugging and installing plug-ins, etc.
- **Side Bar (B)**: Containing different views (like the Explorer) to assist working on projects.
- **Editor (C)**: Working area for editing files. Possible to open multiple editors vertically and horizontally alongside.
- **Minimap (D)**: Code outline that appears on the right side of the editor. Possible to click or drag the shaded area to quickly jump to different sections of the file being edited.
- **Panels (E)**: Different panels displayed below the editor area for output or debug information, errors and warnings, or an integrated terminal.
- **Status Bar (F)**: Information about the opened project and the files being edited.

2. File

2.1. Open File

Click the icon of **GstarLisp** to enter the view. Click **Open Folder** icon, select the directory to load and the **.lsp** and **.dcl** files are filtered out and listed.

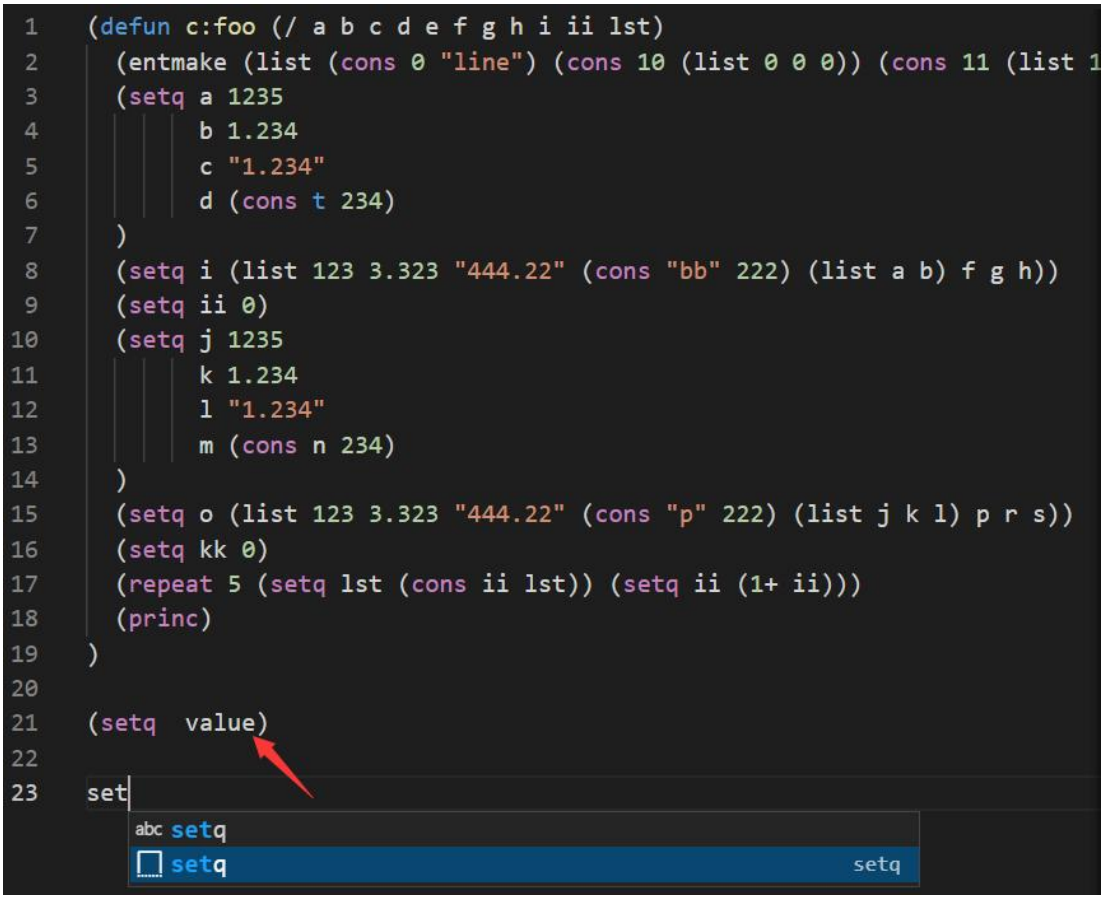
- Click the file(s) in the list to be opened in the editor area on the right.
- Click the **Refresh** to reload the selected directory file(s) if needed.



2.2. Edit File

The opened file(s) can be edited with rich editing functions provided by **GstarLisp**, such as:

- Syntax highlighting
- Smart hints
- Automatic code completion/Code fragment
- Parenthesis matching
- Code folding



3. Debugging

3.1. Debug Mode

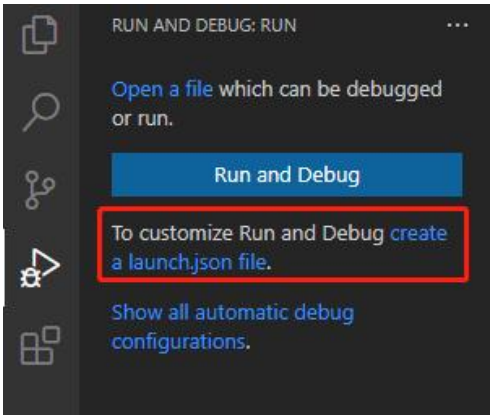
3.1.1. Add Configuration

GstarLisp provides two debug modes: **Attach** mode and **Launch** mode

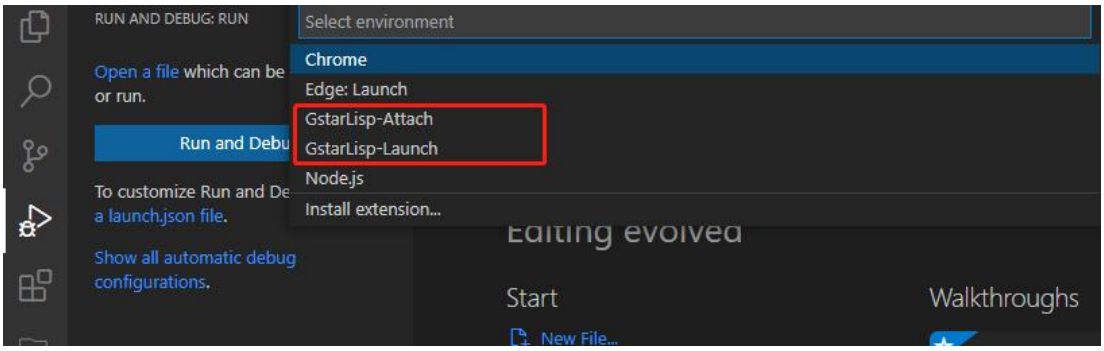
- **Attach** Mode: Attach a debug process to a running CAD process.
- **Launch** Mode: Start a new CAD process by specifying the absolute path of the CAD running file.

The debug mode can be customized by editing the '*launch.json*' file, following steps below:

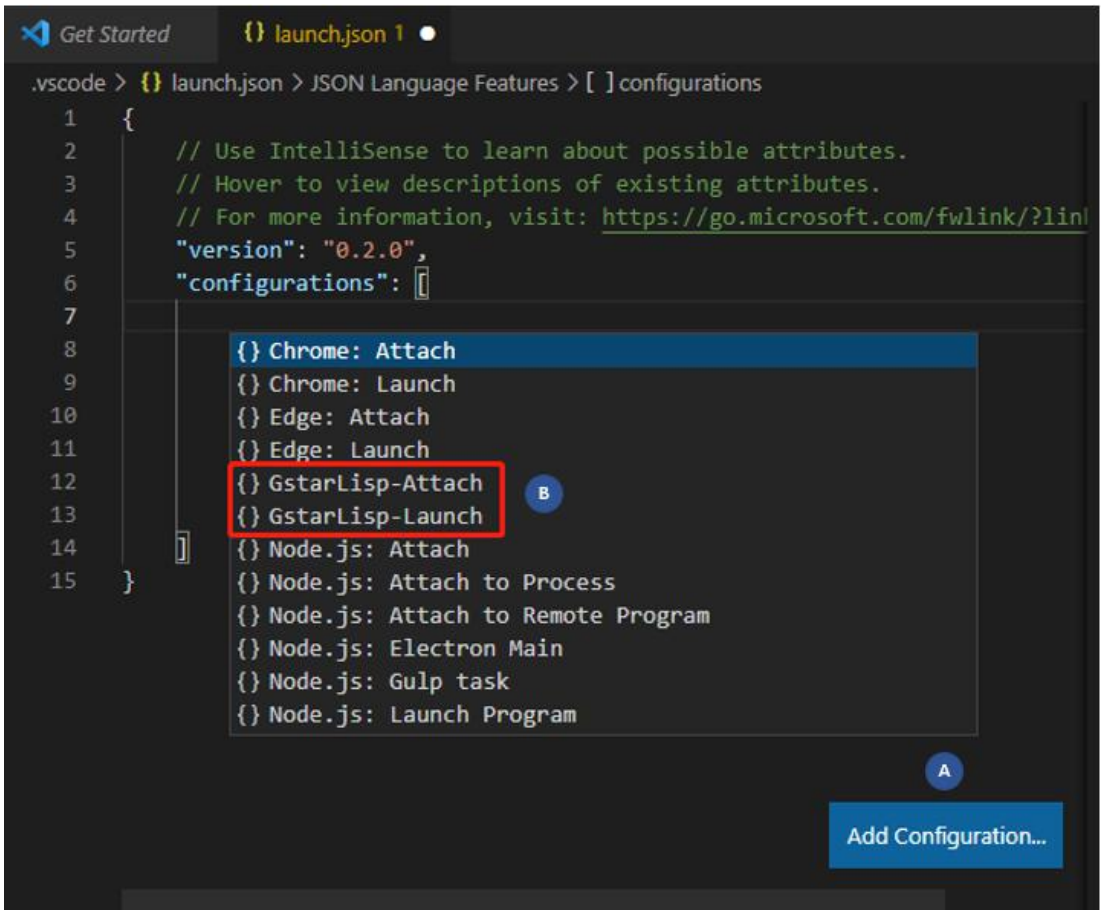
1) Click **create a '*launch.json*' file**.



2) Select a debug mode, **GstarLisp-Attach** or **GstarLisp-Launch**, and add it to '*launch.json*' file.

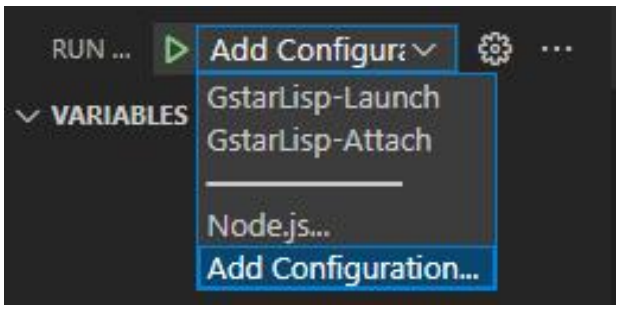


3) Click **Add Configuration** to add another debug mode to '*launch.json*' file.



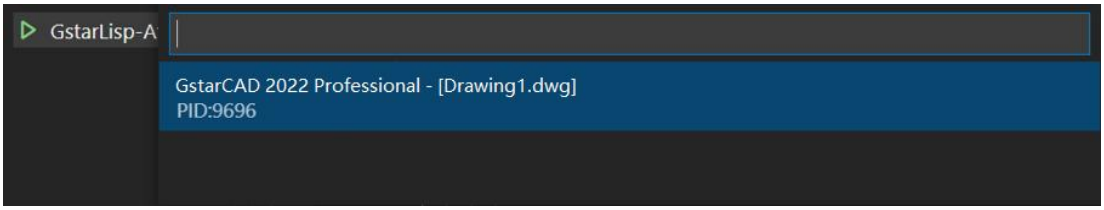
4) Save the modifications to '*launch.json*' file.

5) With the above configuration, it's possible to switch between the two modes during debugging.

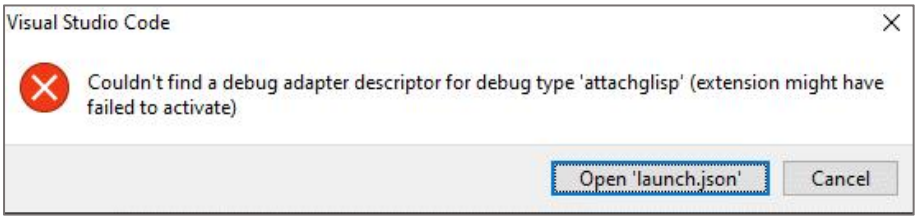


3.1.2. Attach Mode

If **Attach** mode (**GstarLisp-Attach**) is selected, attachable CAD process need to be selected from the list in the debug window, as shown below, and then click **Run** to start debugging.

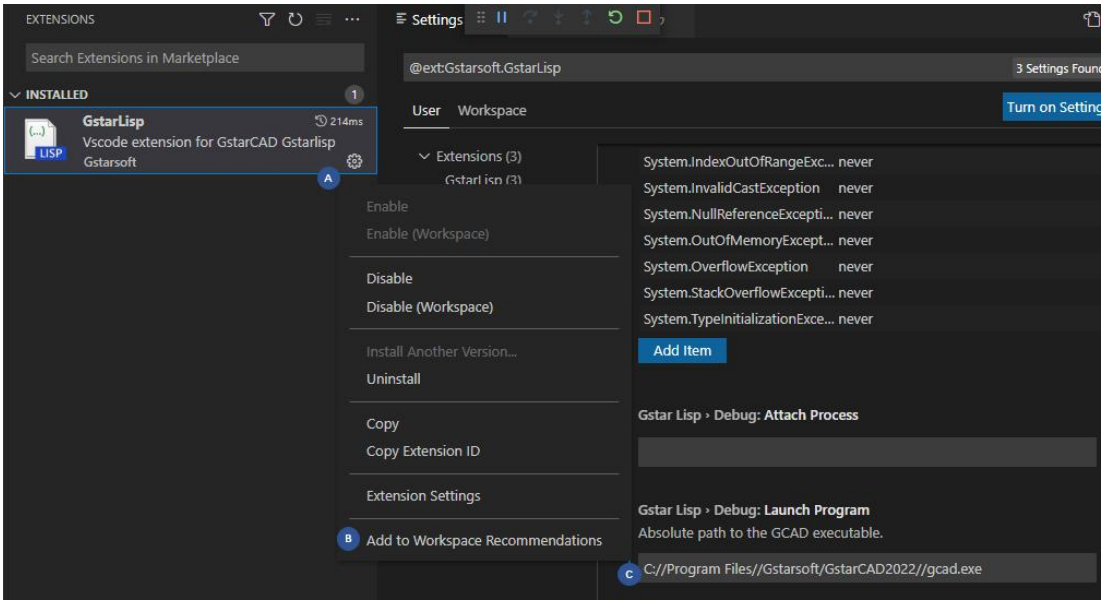


This debug mode is not able to be used if no attachable CAD process is selected. VS Code pops up the following message and prompts to change the configuration of '*launch.json*' file or open the CAD.

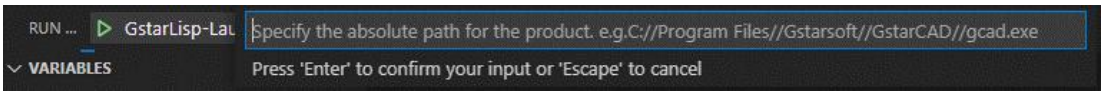


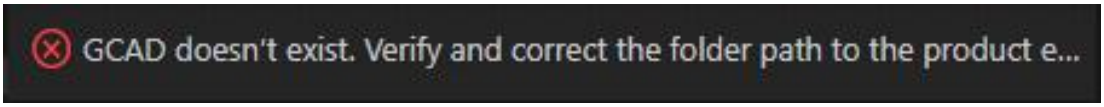
3.1.3. Launch Mode

If **Launch** mode (**GstarLisp-Launch**) is selected, the absolute path of CAD running file (path format: e.g. '*C:\Program files\Gstarsoft\GstarCAD2026\gcad.exe*') needs to be specified in the extension settings of **GstarLisp**, as shown below. After the setting is completed, click **Run** to start a new CAD process for debugging.

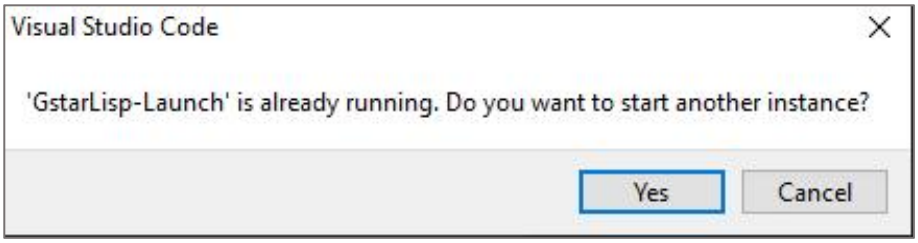


If the absolute path of CAD running file is not specified, a message is displayed in the debug window to prompt specifying the path.





If there is already a running CAD process on the specified path, the following message is displayed when debugging (early versions of VS Code (before 1.57.1) do not have this function). Select **Yes** to start another GstarCAD instance, or select **Cancel** to close the window.



3.2. Side Bar

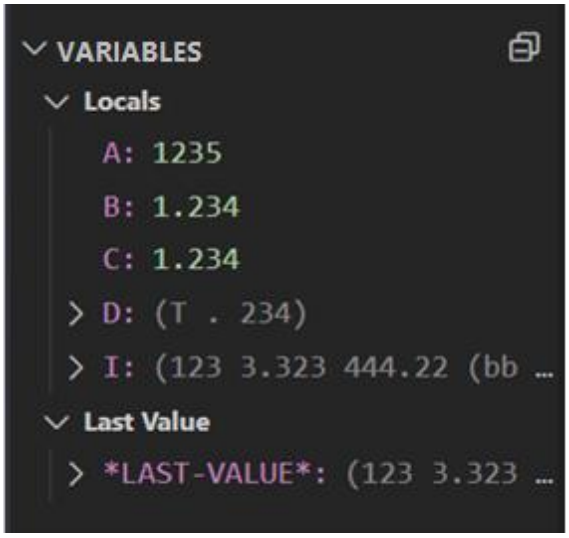
3.2.1. Variable

Locals: Display variable values of the executed code by hovering over their source in the editor.

Last Value: Display the latest value variable.

NOTE:

- When the breakpoint is hit for the first time, the value of **Last Value** is not displayed.
- In **GstarLisp**, **Last Value** displays the latest value of variable while **AutoLisp** displays the latest value/result of variable/expression.



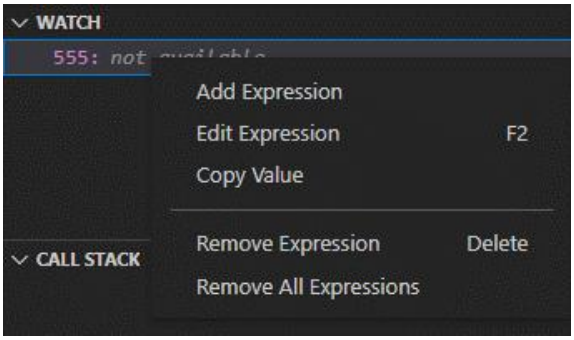
3.2.2. Watch

Watch the real-time value change of specified variables and expressions.

Add the variables or expressions to the watching list by clicking  button. When debugging, the values of variables and expressions in the watching list are real-time updated and displayed.

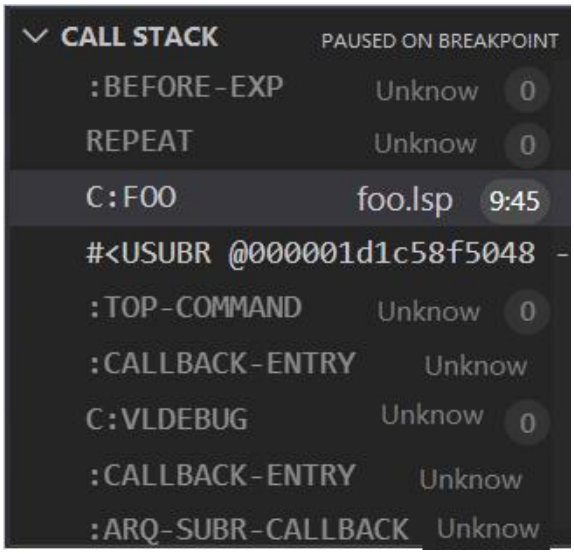
To remove variables or expressions from watching list please  click button.

It is also possible to add, edit and remove an expression from context menu by right-clicking **WATCH**.



3.2.3. Call Stack

In **GstarLisp**, **CALL STACK** mainly displays the names of the called functions. It is possible to jump to the line where a function is located by double-clicking the function name.



3.2.4. Breakpoint

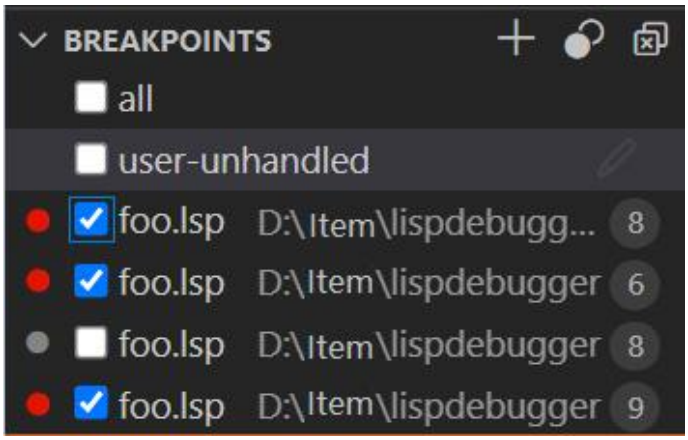
When the program runs to the line where a breakpoint is located, it will be interrupted (NOTE: When the program is interrupted, the line where the breakpoint is located is not run yet).

1) Set breakpoint: select the line of code where a breakpoint is set

- In the left area, click line number to set a breakpoint, click again to cancel the breakpoint.
- In the left area, right-click and select **Add Breakpoint** from the pop menu.
- Press **F9** to set a breakpoint, press **F9** again to cancel the breakpoint.

2) Delete/disable breakpoint:

- Right-click the active breakpoint and select **Delete Breakpoint** or **Disable Breakpoint** from the pop menu.
- Click to delete or disable the breakpoints at the side bar.
- Delete all breakpoints at the side bar.

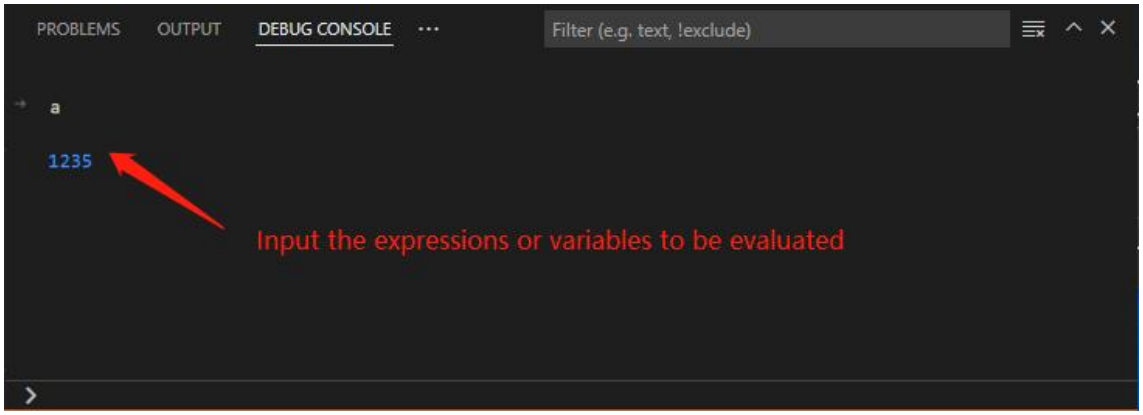


3.3. Debug Console

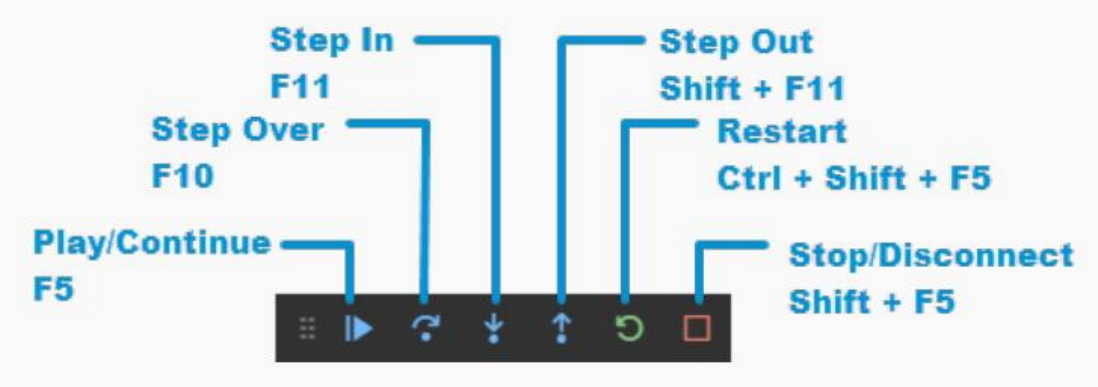
Variables or expressions to be evaluated can be entered below the console.

- When inputting a variable or an expression that exists in the code, the value of the variable or expression is displayed.
- If the input variable or expression does not exist, the name of input variable or expression is displayed.

  In addition, debug console also displays log information to help locate problems.



3.4. Debug Toolbar



3.4.1. Play/Continue

Play/Continue/(F5): Jump to the next breakpoint.

3.4.2. Step Over

Step Over (F10): Execute the current line of code and jump to the next line. It does not stop even the current line contains a function call (execute the whole sub-function as a step).

In **GstarLisp**, press **F10** to execute the line and jump to the next line, while in **AutoLisp**, press **F10** to execute the line, and then press **F10** again to jump to the next line.

3.4.3. Step In

Step In (F11): Step through the line of code. If the current line contains a function call, execute the first line of the called function (enter the sub-function and continue stepping in).

3.4.4. Step Out

Step Out (Shift+F11): When stepping in a sub-function and stepping out, execute the unexecuted section of the function.

3.4.5. Restart

Restart (Ctrl+Shift+F5)

Under **Attach** Mode: Restart current debugging without ending the current CAD process.

Under **Launch** Mode: Terminate the current debugging, end the current CAD process (without asking whether to save the drawing) and restart a new CAD process for debugging.

3.4.6. Stop/Disconnect

Stop/Disconnect (Shift+F5):

Under **Attach** Mode: Stop the current debugging without ending the current CAD process.

Under **Launch** Mode: Stop the current debugging, and end the current CAD process (without asking whether to save the drawing).

4. Copyright

GstarCAD 2027

https://www.gstarcad.net

Gstarsoft