



浩辰CAD Linux版 2024

LISP开发指南

苏州浩辰软件股份有限公司

目录

1. 前言	2
2. 环境配置	2
2.1 安装 Visual Studio Code	2
2.2 安装 GstarLisp 插件	2
2.3 Visual Studio Code 工作界面	3
3. 文件	4
3.1 打开文件	4
3.2 编辑文件	5
4. 调试	6
4.1 调试模式	6
4.1.1 添加配置	6
4.1.2 Attach 模式	8
4.1.3 Launch 模式	8
4.2 调试侧边栏	9
4.2.1 变量	9
4.2.2 监视	10
4.2.3 调用堆栈	10
4.2.4 断点	11
4.3 调试控制台	12
4.4 调试工具栏	12
4.4.1 继续	12
4.4.2 单步跳过	13
4.4.3 单步调试	13
4.4.4 单步跳出	13
4.4.5 重启	13
4.4.6 停止	13
5. 版权声明	13

1. 前言

浩辰 CAD 平台软件（Linux 版）自带 Lisp 语言解释器，默认支持 Lisp 语言，用户可以随时使用 Lisp 语句，它是被解释执行的，任何一个语句键入后就能马上执行，所以它对于交互式的程序开发是非常便利的。

2. 环境配置

2.1 安装 Visual Studio Code

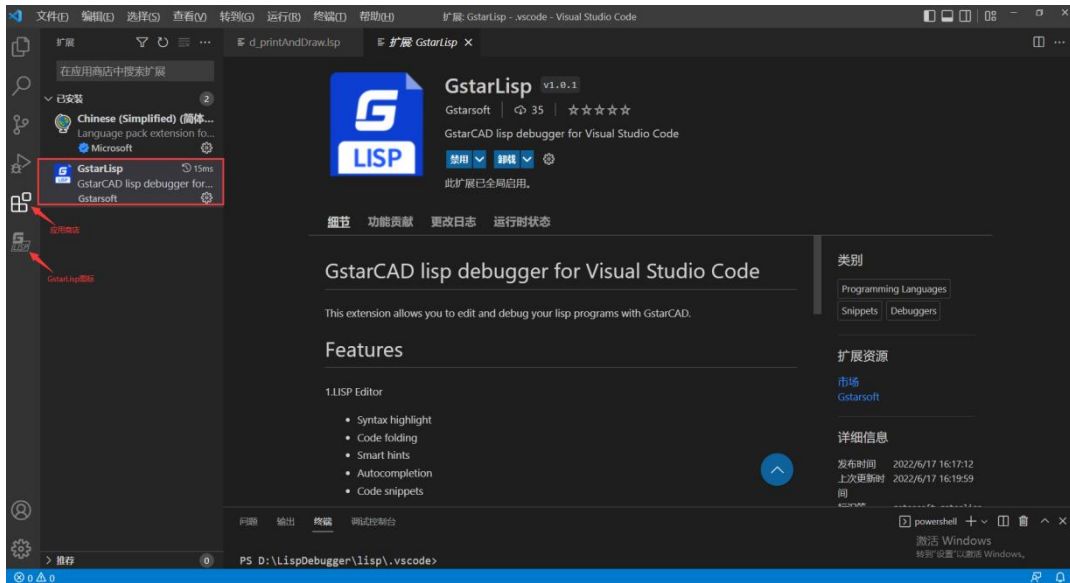
Visual Studio Code（简称：VS Code）是一款由微软开发且跨平台的免费源代码编辑器。利用 VS Code 所提供的功能，用户可以在浩辰 CAD 中调试代码。通过在浩辰 CAD 命令行中输入 VLIDE 或 VLISP，调起 VS Code。如果用户的电脑上没有安装 VS Code，请先在 VS Code 官网下载并安装 VS Code。

2.2 安装 GstarLisp 插件

通过浩辰 CAD 启动 VS Code 时，将会默认帮用户安装 GstarLisp 插件。当 GstarLisp 插件有新版本时，在用户下次通过浩辰 CAD 启动 VS Code 时将会自动更新。

另外，用户也可以选择手动安装 GstarLisp 插件。通过在 VS Code 应用商店中搜索“GstarCAD”、“CAD”或“LISP”，选择“GstarLisp”再点击安装。

插件安装成功后，活动栏将显示 GstarLisp 插件的图标，用户还可以查看 GstarLisp 插件的细节展示，包括插件名称、简介、基本特性、新功能、如何安装、如何使用、授权许可信息、变更日志等信息，如下图所示。

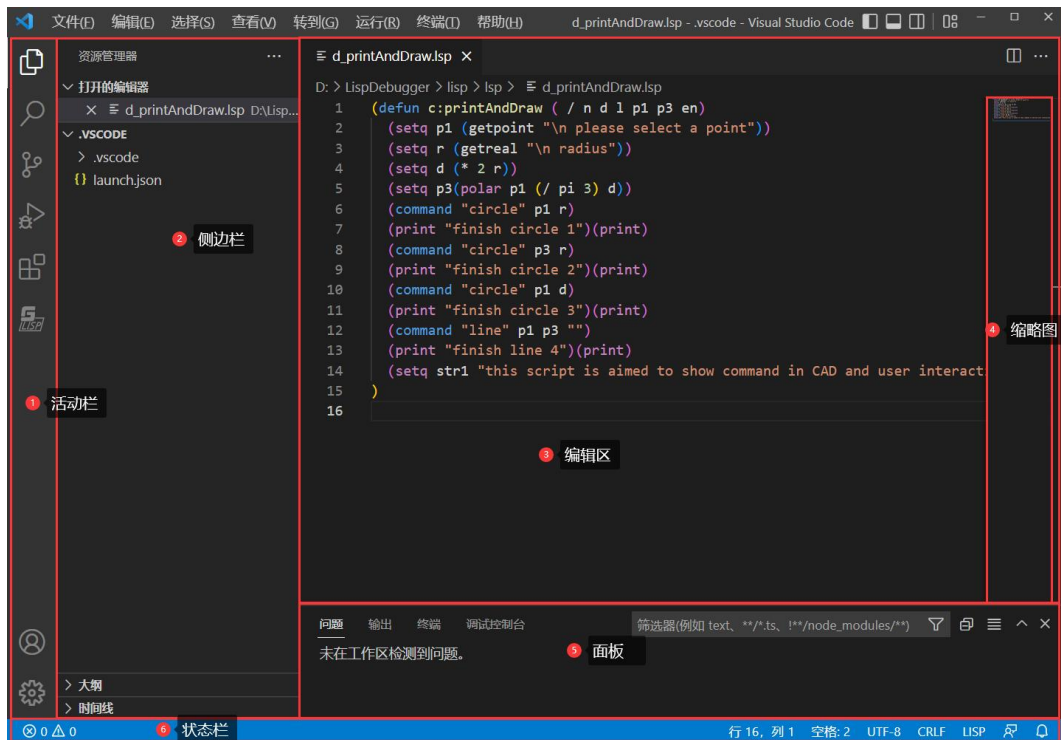


2.3 Visual Studio Code 工作界面

VS Code 和 GstarLisp 插件都安装完成后，在浩辰 CAD 命令行输入 VLIDE 或 VLISP，将启动 VS Code。

```
命令:VLISP
正在安装 GstarCAD Visual LISP Editor, 请稍等。
命令:GstarCAD Visual LISP Editor 已安装, Visual Studio Code正在启动。
```

下图为 VS Code 工作界面展示。



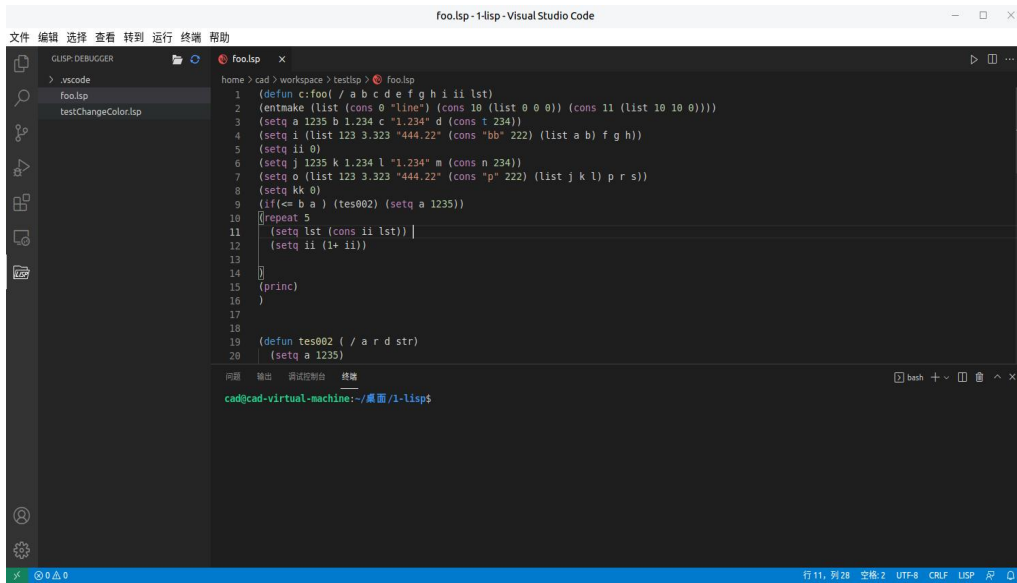
- **活动栏**：位于最左侧，允许用户在视图之间切换，可以选择打开文件、调试或者安装插件等。
- **侧边栏**：包含不同的视图（如资源管理器），可在处理项目时为用户提供帮助。
- **编辑区**：用于编辑文件的主要区域。用户可以根据需要垂直和水平并排打开任意数量的编辑器。
- **缩略图**：即代码大纲，显示在编辑器的右侧，用户可以单击或拖动阴影区域以快速跳转到文件的不同部分。
- **面板**：用户可以在编辑器区域下方显示不同的面板，用于输出或调试信息、错误和警告或集成终端。
- **状态栏**：有关打开的项目和用户编辑的文件的信息。

3. 文件

3.1 打开文件

点击 GstarLisp 插件图标，进入插件视图。点击“打开文件夹”，选择加载目录，将会筛选出.lsp 文件，如下图所示。

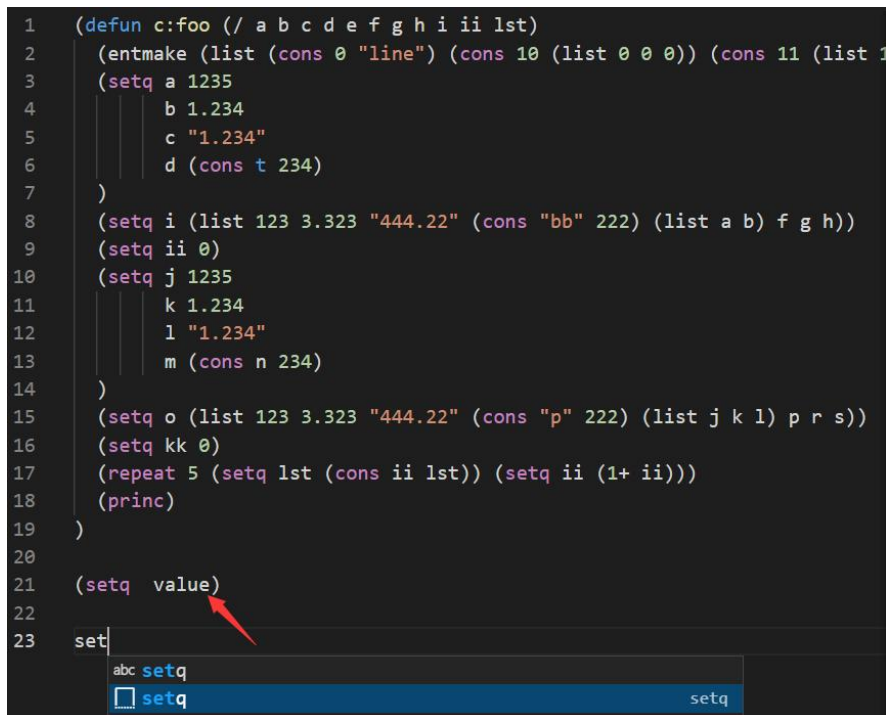
- 单击列表中的文件，则右侧编辑区打开文件。
- 支持刷新，点击刷新按钮可以重新加载已选择的目录文件。



3.2 编辑文件

打开的文件，支持编辑。GstarLisp 插件为用户提供了丰富的编辑功能，如：

- 语法高亮
- 智能输入提示
- 代码自动补全&代码片段
- 括号匹配
- 代码折叠



4. 调试

4.1 调试模式

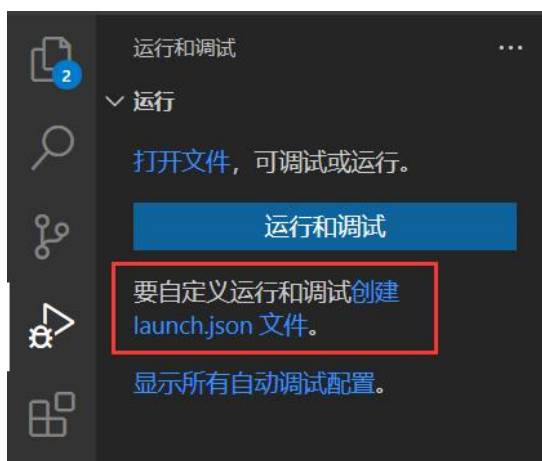
4.1.1 添加配置

GstarLisp 插件为用户提供了两种调试模式：Attach 模式和 Launch 模式。

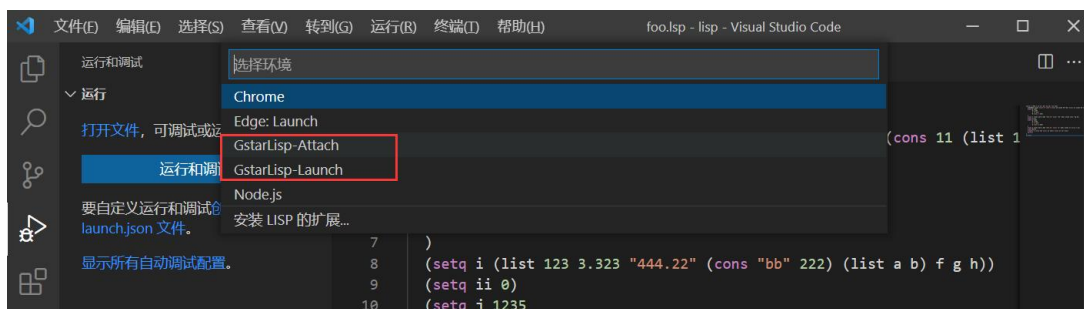
- Attach 模式：将调试会话附着到正在运行的 CAD 会话。
- Launch 模式：通过指定 CAD 可执行文件的绝对路径，启动新的 CAD 会话。

用户可以通过编辑 launch.json 文件自定义调试模式。步骤如下：

Setp1: 点击“创建 launch.json 文件”；



Setp2: 选择调试模式 GstarLisp-Attach 或 GstarLisp-Launch，添加到 launch.json 文件中；



Setp3: 点击“添加配置”，将另一个调试模式也加到 launch.json 文件中；

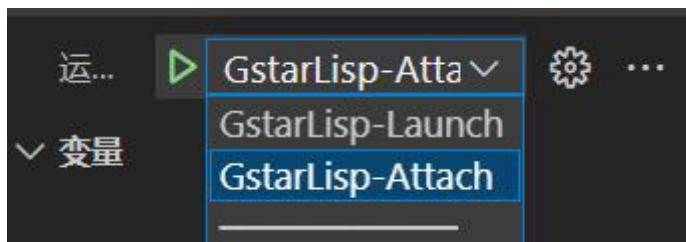
```
.vscode > {} launch.json > JSON 语言功能 > [ ] configurations
1  {
2      // 使用 IntelliSense 了解相关属性。
3      // 悬停以查看现有属性的描述。
4      // 欲了解更多信息，请访问：https://go.microsoft.com/fwlink/?linkid=830
5      "version": "0.2.0",
6      "configurations": [
7
8          {} Chrome: Attach
9          {} Chrome: Launch
10         {} Edge: Attach
11         {} Edge: Launch
12         {} GstarLisp-Attach ②
13         {} GstarLisp-Launch
14         {} Node.js: Attach
15         {} Node.js: Attach to Process
16         {} Node.js: Attach to Remote Program
17         {} Node.js: Electron Main
18         {} Node.js: Gulp task
19         {} Node.js: Launch Program
20         "name": "GstarLisp-Attach"
21     ]
22 }
23
24 }
```

① 添加配置...

Step4: 保存 launch.json 文件的修改;

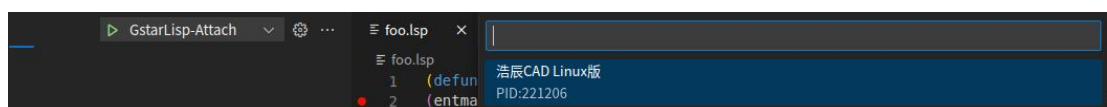
```
{
  // 使用 IntelliSense 了解相关属性。
  // 悬停以查看现有属性的描述。
  // 欲了解更多信息，请访问：https://go.microsoft.com/fwlink/?linkid=830
  "version": "0.2.0",
  "configurations": [
    {
      "type": "launchlisp",
      "request": "launch",
      "name": "GstarLisp-Launch"
    },
    {
      "type": "attachlisp",
      "request": "attach",
      "name": "GstarLisp-Attach"
    }
  ]
}
```

Step5: 现在，用户就可以在调试时选择不同的调试模式。

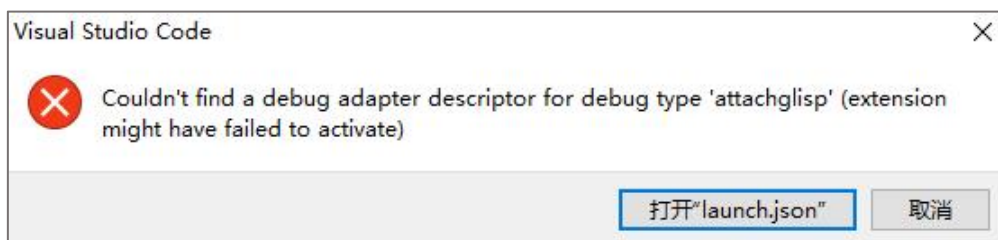


4.1.2 Attach 模式

若用户选择 Attach 模式，需要在调试窗口的列表中选择用户要附着的 CAD 进程，如下图所示，然后点击运行，即可进行调试。

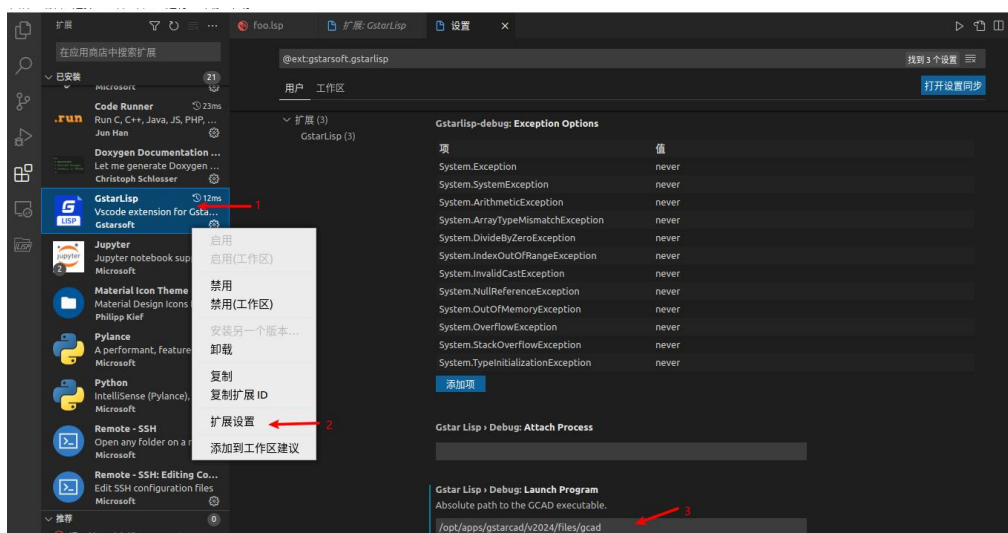


如果用户没有打开任何可附着的进程，则无法使用该调试模式，VS Code 会提示如下弹窗，需要用户更改 launch.json 配置或者打开 CAD。

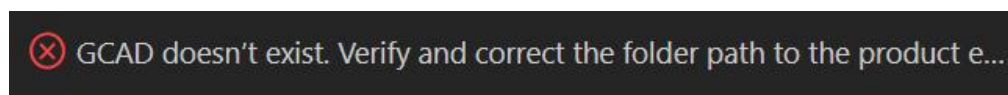


4.1.3 Launch 模式

如果用户选择 Launch 模式，需要在插件的扩展设置中，指定 CAD 可执行文件的绝对路径(路径格式: /opt/apps/gstarcad/v2024/files/gcad)，如下图所示，然后点击运行，这时会启动新的 CAD 进程，进行调试。



如果用户没有设置 CAD 可执行文件的绝对路径，则调试窗口会提示用户设置路径。



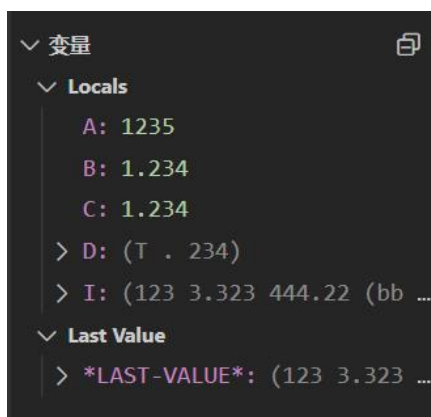
4.2 调试侧边栏

4.2.1 变量

Locals: 显示作用域内已执行代码的变量值。当鼠标悬停到变量上时会显示该变量的值。


Last Value: 显示最近一次变化的变量或表达式的值。

- 第一次命中断点，Last Value 值不显示。
- GstarLisp 中，Last Value 显示的是最近一次变量的值；而 AutoLisp 中显示的最近一次变量或表达式的值。

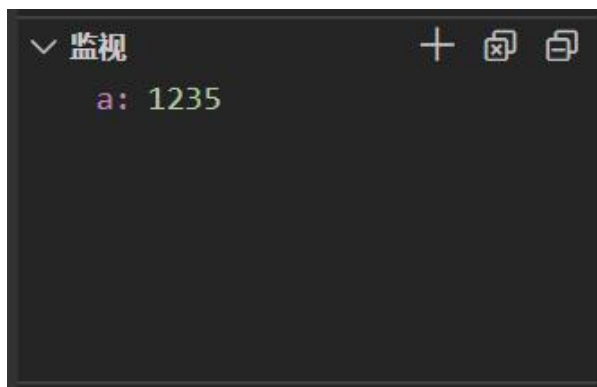


4.2.2 监视

查看和计算变量与表达式。

通过  可以添加要监控的变量或表达式，调试过程中实时更新该变量或表达式的值。

通过  可以删除监视的所有变量或表达式。



右键可以选择添加、编辑、删除单个表达式。



4.2.3 调用堆栈

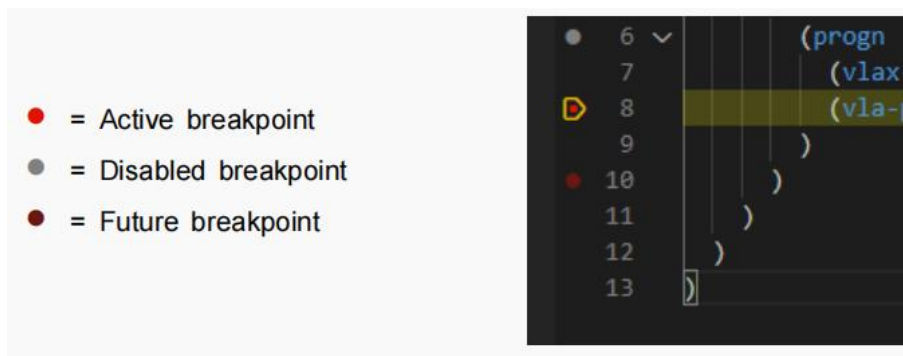
在程序执行过程中存储函数调用信息的数据结构。

GstarLisp 中，调用堆栈主要显示调用的函数名。双击函数名可以跳转到该函数所在的行。



4.2.4 断点

程序执行到断点所在的行，就会中断执行（注意：程序中断后，断点所在的行还没有执行）。

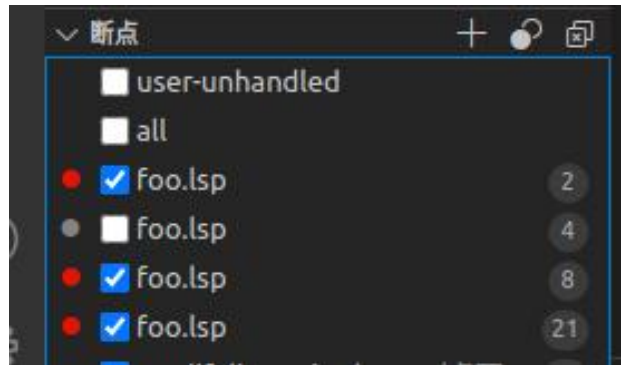


(1) 设置断点：选定要设置断点的代码行

- 在左边区域，单击鼠标左键即可设置断点，再次单击取消断点；
- 在左边区域，单击鼠标右键，在弹出的菜单中选择添加断点；
- 按 F9 设置断点，再次单击取消断点。

(2) 删除/禁用断点

- 选中激活的断点单击鼠标右键，在弹出的菜单中选择删除或禁用断点；
- 侧边栏点击删除或禁用断点；
- 侧边栏删除全部断点。



4.3 调试控制台

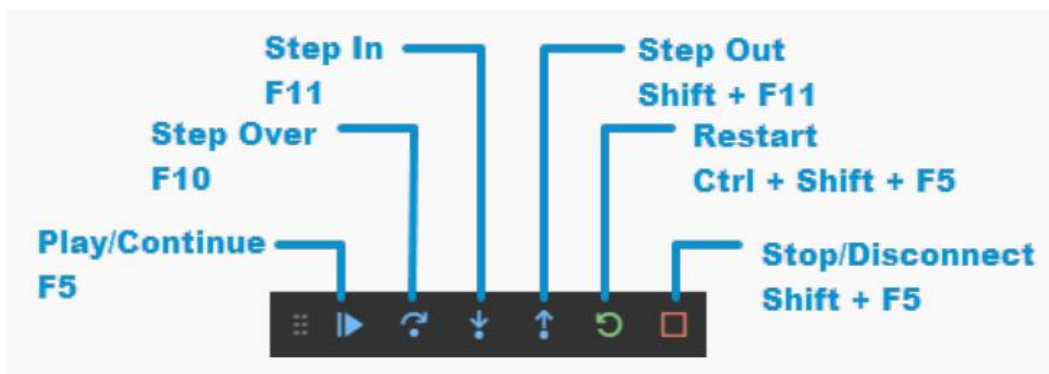
可以在控制台下方输入要评估的表达式或变量。

- 输入代码中存在的变量或表达式，则显示变量或表达式的值。
- 若输入的变量不存在，则显示输入内容。

另外，调试控制台还显示日志信息，帮助用户定位问题。



4.4 调试工具栏



4.4.1 继续

继续 (F5): 跳到下一个断点。

4.4.2 单步跳过

单步跳过/step over (F10): 执行当前行代码并跳到下一行。不会因为当前行包含一个函数调用而停止 (将子函数整个作为一步执行完)。

GstarLisp 中按 F10 则执行该行同时跳到下一行, 而 AutoLisp 中, 按 F10 则执行该行, 再按 F10 再跳到下一行。

```
6 (setq j 1235 k 1.234 l "1.234" m (cons n 234))
```

4.4.3 单步调试

单步调试/step in (F11): 单步执行该行代码。如果当前行包含一个函数调用, 则执行被调用函数的第一行 (进入子函数并继续单步执行)。

```
7 (setq o (list 123 3.323 "444.22" (cons "p" 222) (list j k l) p r s))
```

4.4.4 单步跳出

单步调试/step out (Shift+F11): 当单步执行到子函数内, 单步跳出时, 执行该函数未执行的部分。

4.4.5 重启

Attach 模式: 当前调试不会终止, 在当前 CAD 进程重新开始调试。

Launch 模式: 终止当前调试会话, 结束当前 CAD 进程 (不询问图纸是否保存), 再重新拉起新的 CAD 进程进行调试。

4.4.6 停止

Attach 模式: 停止当前调试会话, 不结束 CAD 进程。

Launch 模式: 停止当前调试会话, 并结束 CAD 进程 (不询问图纸是否保存)。

5. 版权声明

版权所有: 苏州浩辰软件股份有限公司。

使用许可：允许复制、引用本文档的任何部分。未经许可，不得更改本文档的任何部分。在复制、引用时，请务必保留本声明，否则将追究法律责任。