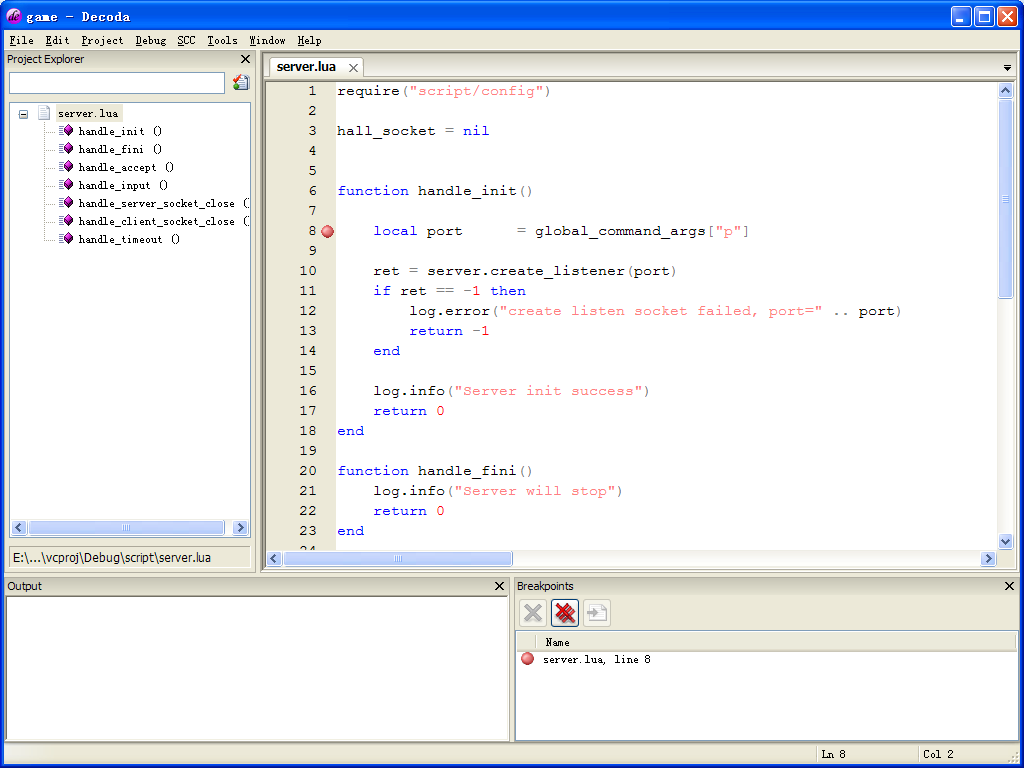
|  |
| --- |
| **使用Decoda工具调试Lua程序** |

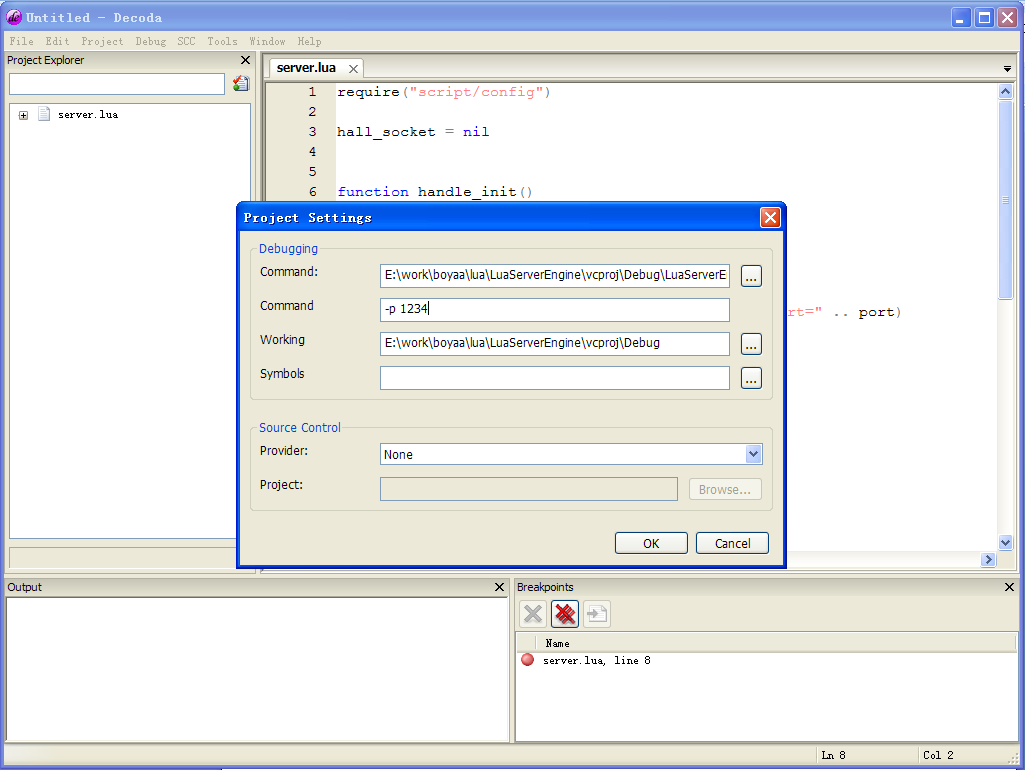
Decoda是windows上调试lua的工具，类似于我们常用的Microsoft Visual Studio调试，支持断点，单步，跳进跳出，观察变量值，功能很强大，使用此工具，我们可以不用再打一堆的调试日志。

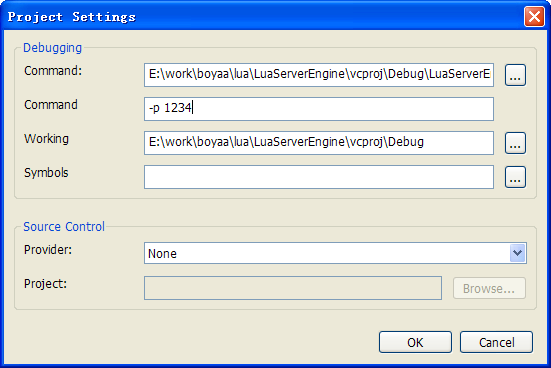


## 调试方法一

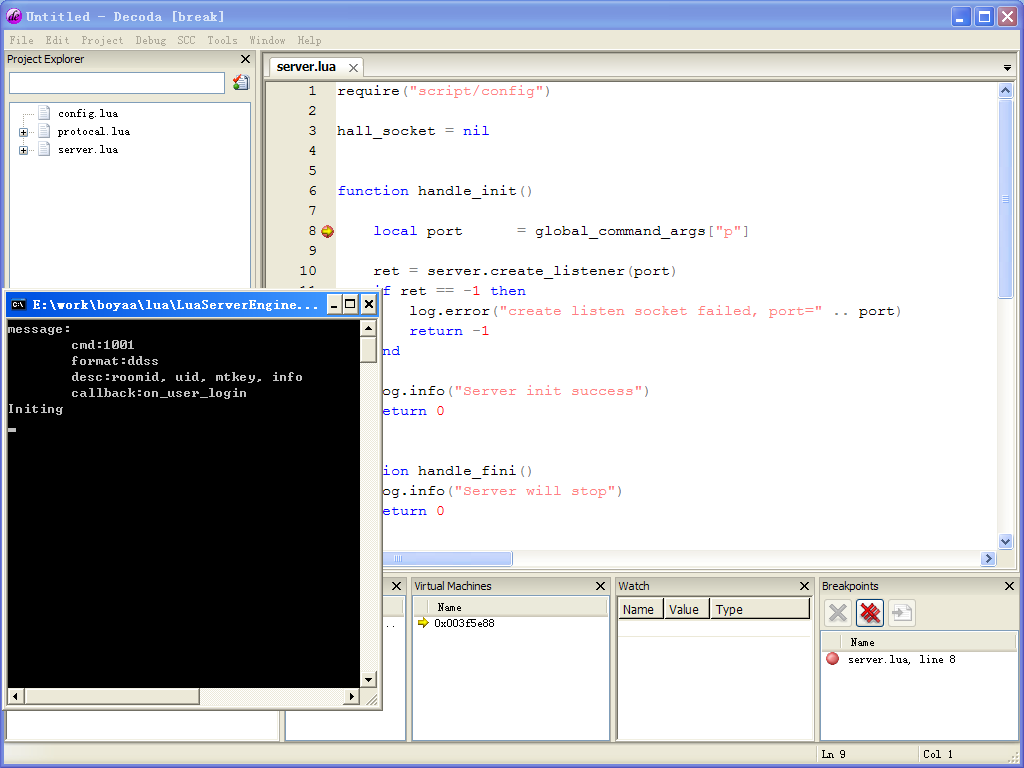
指定一个宿主程序，由宿主程序来加载lua文件进行调试。

1. 在这里演示一个server.lua文件的调试方法，该文件在server启动时会被加载，并调用其中的handle\_init函数，我们在handle\_init第一行设置一个断点（在断点行处按F9键或点击编辑区左边行号处），如上图，我们在第8行处设置了一断点

2.设置完断点后，按F5键开始运行，第一次运行时，会弹出一个project settings对话框



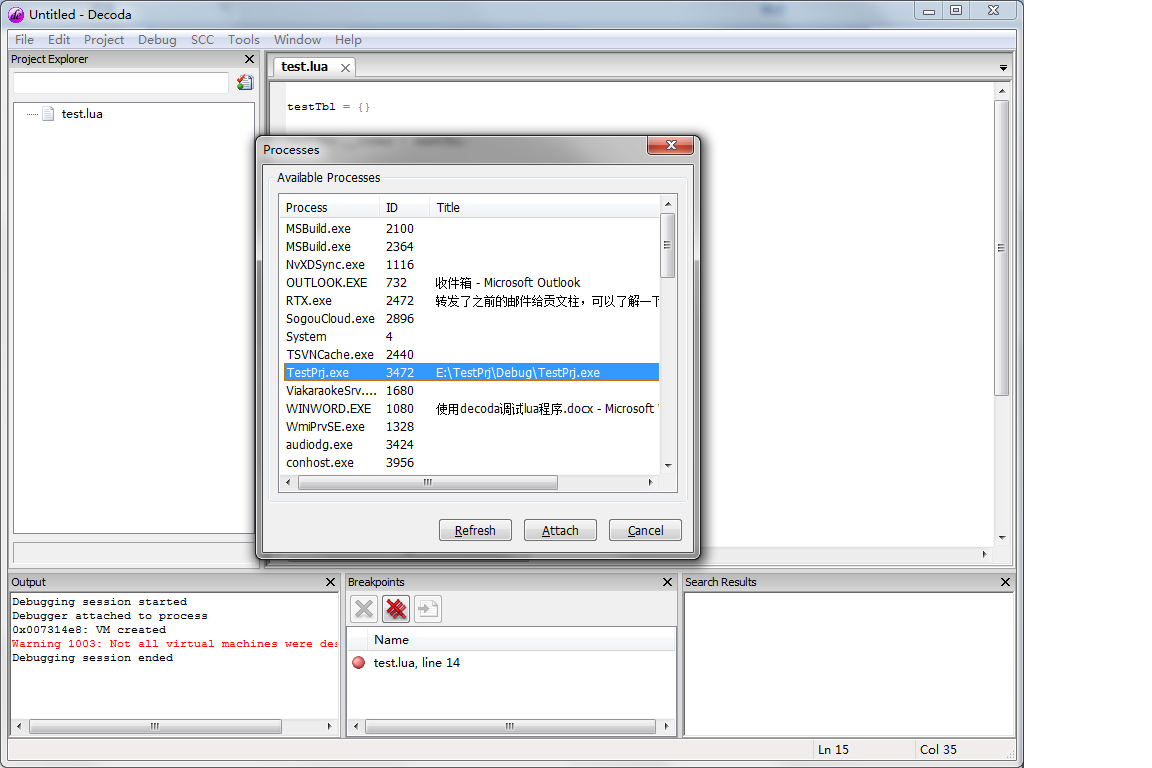
3. 第一行中选择宿主程序exe，第二行指定程序的命令行参数，如果没有就留空，第三个是工作目录（如果工作目录就是exe所在目录，可以选择默认），第四个是宿主exe对应的pdb（如exe同pdb在同一目录下，可以留空），OK之后，就开始运行exe，这时程序运行到第8行断点处。



## 调试方法二：

Decoda 注入宿主程序进行调试。

1. 在宿主程序，需要中断的地方，打一个软断点，比如用getchar() 阻塞，让程序停在需要中断的地方。
2. 直接启动宿主程序。
3. 打开Decoda，打开需要中断的lua文件，打好断点。
4. 选择菜单 Debug->Processes 打开一个查看当前运行进程的对话框，选择宿主程序运行的进程，点击Attach。



1. 继续运行中断的宿主进程，如用getchar()阻塞，则输入一个字符。然后就会停在lua中所打断点处。

## 其他：

1. Debug菜单如图，常用的有：

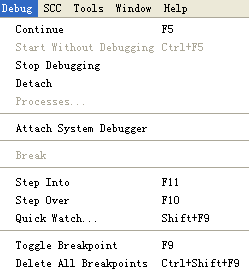
F5继续运行

F9设置断点

F10单步往下运行

F11跳入

Shift+F9观察变量，可将变量加入到watch列表中.



2. 调试环境保存为工程，在菜单project—>save project保存起来，下次调试时直接打开工程，就不要设置宿主程序参数，断点等工作了。

3、改变工程设置，如需改变 step 2 中的工程设置，可以通过菜单 project->settings ，这时也会弹出prject settings 对话框，可以改变当前的工程设置。

4、如output 中出现waring，双击对应的warning 可以打开帮助文档，查看对应信息。

5、目前Decoda 指支持5.1.\* 。