

+ Follow + Vote Flag

0

How to debug MEX-file compiled with MinGW64 and -g flags

Asked by MathWorks Support Team **STAT1** on 1 Sep 2015
 Latest activity Commented on by Lodi.S3 on 13 Dec 2017
 Accepted Answer by MathWorks Support Team **STAT1**

How to debug MEX-file compiled with MinGW64 and -g flags

3 Answers

+ Vote

0

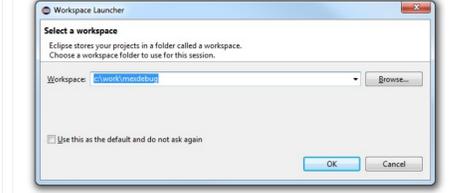
Flag Link

Answer by MathWorks Support Team **STAT1** on 29 Jan 2016

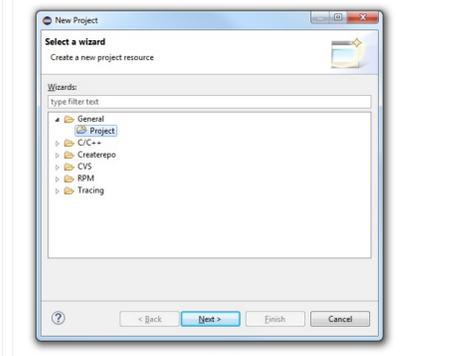
Accepted Answer

Follow the steps below to debug a MEX-file compiled with MingW64 and the -g flag:

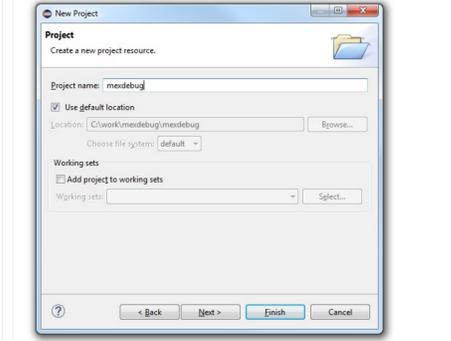
1. Start Eclipse and choose a new workspace.



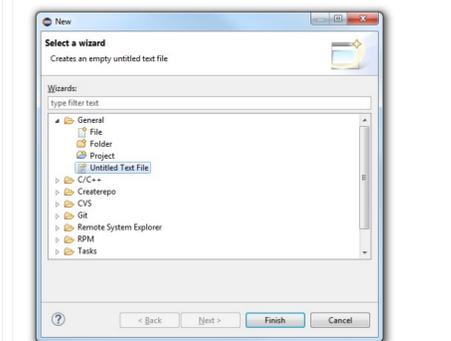
2. Use "File" -> "New" -> "Project" ... and select "General"->"Project"



3. Choose a name for your project and click "Finish"



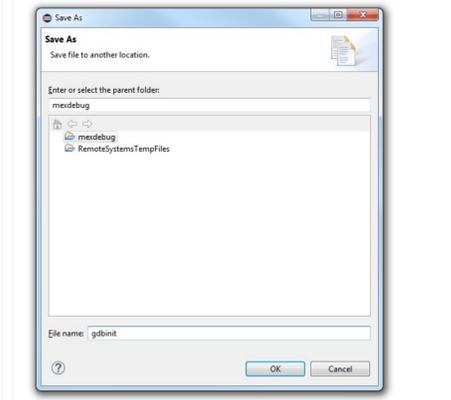
4. Use "File"->"New"->"Other"..." and then "General"-> "Untitled Text File" to create a new text file.



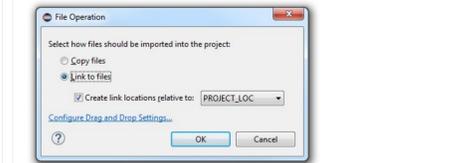
Enter the following in the text file

```
handle SIGSEGV nostop
```

Then save the file as "gdbinit" in "mexdebug" folder as shown in the figure.



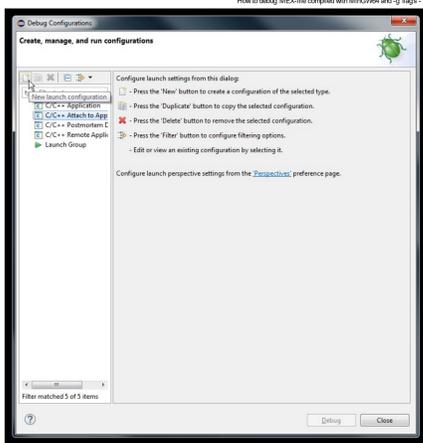
5. Drag the C/Cpp source code of your MEX-file into the project. When asked to copy or link choose "Link to file" so that the source codes stay in their original locations.



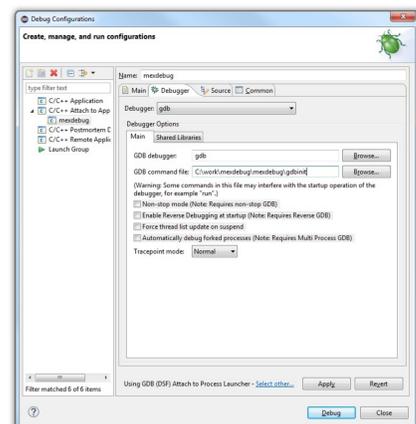
6. Go to "Run" -> "Debug Configurations..." . Select "C/C++ Attach to Application" and select "New" button.

Related Content

- MATLAB Answers
- How can I debug a Fortran MEX file on Windows using Intel, Compaq, or Digital Visual Fortran compilers with Microsoft Visual Stud...
 - 1 Answer
 - How do I debug C MEX-files under UNIX?
 - 1 Answer
 - How do I debug a MATLAB C/C++ MEX-file in Microsoft Visual Studio 2005?
 - 1 Answer
 - Entire Website
 - MATLAB Support Package for ParallaxBASIC Stamp
 - File Exchange
 - Debugging C/C++ S-Functions
 - Blogs
 - Getting Started with MATLAB Support Package for Arduino Hardware
 - Examples
 - Tags
 - mex
 - file
 - mingw64
 - g
 - flags
 - Products
 - MATLAB Compiler
 - Discover what MATLAB® can do for your career.
 - Opportunities for recent engineering grads.
 - » Apply Today

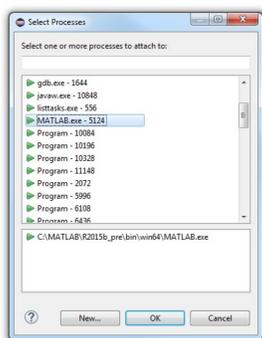


7. On the "Debugger" tab enter the location of the file created under 4 as "GDB command file".

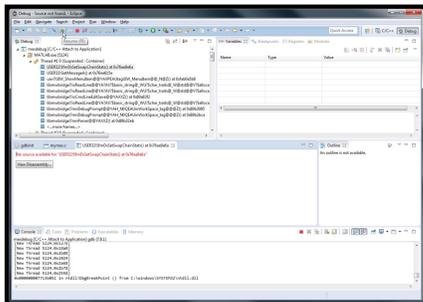


8. Click "Debug"

9. Select MATLAB.exe from the list.



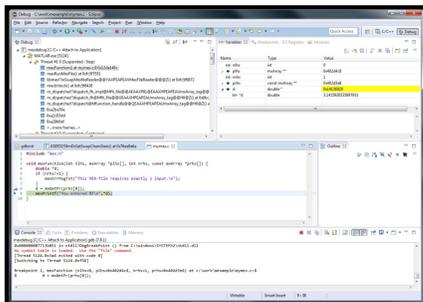
10. The debugger will suspend MATLAB. Use "Resume" button to continue running MATLAB.



11. Open the C/Cpp code in Eclipse and place a breakpoint.

12. Run the MEX-file in MATLAB

13. When the breakpoint is hit you will stop in the debugger and can for example step through the code and inspect variables.



3 Comments

[Martin Cacan](#) on 30 May 2017

Is this the new preferred method to debug C code mex'd to run in Matlab? Other documentation indicates using Microsoft Visual Studio 2012 Professional. However, Microsoft is no longer supporting VS 2012, and hence not selling keys to the professional edition any longer.

[Liang Yin](#) on 16 Sep 2017

This answer, as well as the comments to the other two answers, were super helpful, thank you very much! However, how did everyone deal with the large amount of errors that popped out.

[LodL93](#) on 13 Dec 2017

Unfortunately, this is not working for me. I don't see the MATLAB Process in the process list to select from. I use MATLAB R2017a 64bit und Eclipse Oxygen. I wonder if this can be due to 32/64bit mismatches, especially between the GDB and the MATLAB-Process, since I have both MinGW (32bit) and MinGW-w64 installed on my computer. I assume a GDB 64bit is required to attach to a 64 bit Application, but don't know how to tell eclipse which GDB to use. Are there other possible reasons why I can't select MATLAB.exe to attach to, and if not, how can I tell Eclipse to use the MinGW64 GDB? Thank you very much.

[Comment on this Answer](#)



[Vote](#)
0
[Flag](#) [Link](#)
Answer by [Chuong Nguyen](#) on 10 Jun 2017

Thank you very much, but for my case, when the window "select process" open, I can't see the matlab.exe process in the windows?, can you help me to solve the problem?

1 Comment

[Lithe](#) on 13 Jun 2017

I just tried applying this procedure and ran into two issues. Fortunately, both were small and easily resolved.

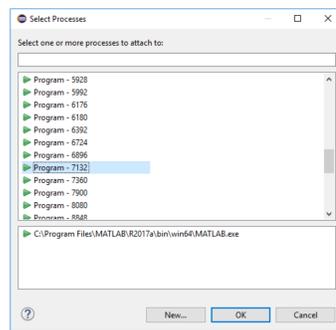
1. the version of gdb included with the latest version of TDM-GCC MinGW has a compatibility problem with Eclipse. For more information, see https://bugs.eclipse.org/bugs/show_bug.cgi?id=474311

I was able to solve this problem by using an earlier version. By trial and error, I found that the following version works:

tdm64-gcc-4.7.1-3.exe

(Note, this issue has nothing to do with MATLAB, per se).

2. As noted in the previous message, the task does not come up as MATLAB.com. The correct task is one of the generically named tasks called "Program" (at least in windows 10). But by clicking on the name, you can quickly identify the correct task (see attachment).



With these changes to the procedure, it works great.

[Comment on this Answer](#)



Vote

0

Flag

Link

Answer by Chuong Nguyen on 10 Jun 2017

and when I detect "Program 15920" is the link to matlab, I press "ok" but again, Eclipse show the error:

```
Error in final launch sequence Failure to attach to process: MATLAB.exe [15920] Error: Failed to execute MI command: -target-attach 15920 Error message from debugger back end: Can't attach to process.
```

can you help me to solve this problem

[1 Comment](#)



Anton Drovjannikov on 21 Aug 2017

Maybe, this is because you use not compatible with matlab debugging version of gdb. I've got the same problem and solution was to install mingw64 (<https://sourceforge.net/projects/mingw64/>), version 4.9.1 (it uses GNU gdb (GDB) 7.8.1 ver., Architecture x86_64, threads win32, exception dwarf, Build version 3. And also 64bit Matlab and eclipse(oxygen, neon, luna)

[Comment on this Answer](#)

Answer this question

Body

[Bold](#)[Italic](#)[Monospaced](#)[Bulleted list](#)[Numbered list](#)[Code](#)[Hyperlink](#)[Image](#)[Attach](#)[File](#)[Help](#)

Preview [Disable](#)

Follow activity on this question ([change notification settings](#))

mathworks.com

© 1994-2018 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.