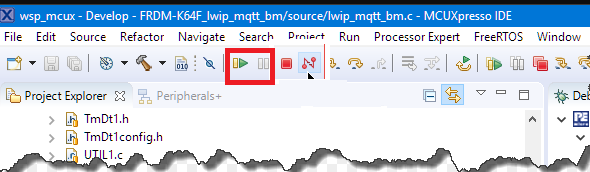
<http://www.vogella.com/tutorials/EclipseDebugging/article.html>

Shortcut key to step through debugging

* Step Into (F5): Executes the currently selected line and goes to the next line in your program.
  + If the selected line is a method call the debugger steps into the associated code.
* Step over (F6) : It executes a method without stepping into it in the debugger.
* Step Return (F7): This finishes the execution of the current method and returns to the caller of this method.
* Resume (F8) : Resume the execution of the program code until is reaches the next breakpoint or watch point.

Suspend: If resume takes too long to reach next brake-point you can suspend that and again resume later.

Resume: execute until next break point

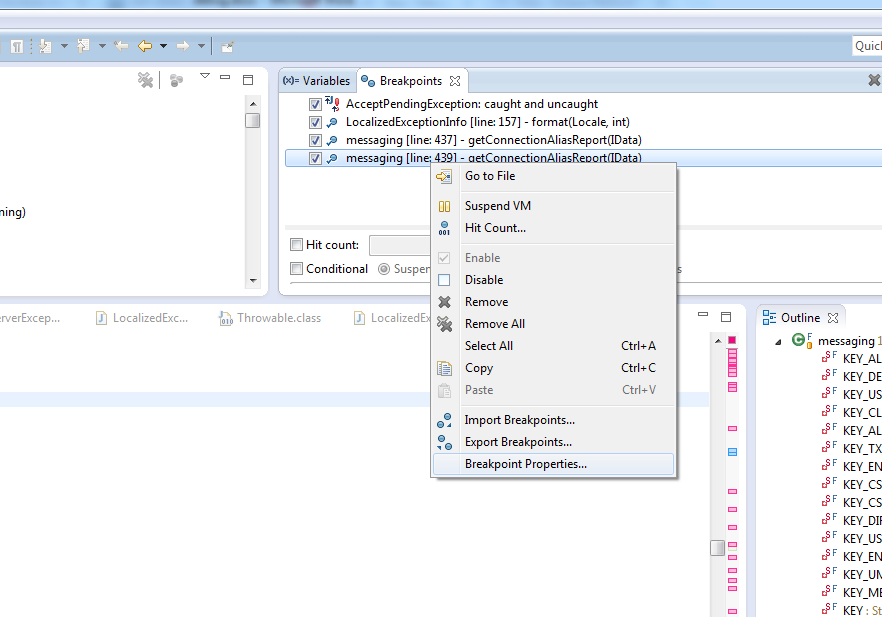


The call stack shows the parts of the program which are currently executed and how they relate to each other. The current stack is displayed in the Debug view.

The Variables view displays fields and local variables from the current executing stack.

Breakpoints view allows you to delete, deactivate or skip breakpoints and watchpoints. You can also modify their properties.

After setting a breakpoint you can select the properties of the breakpoint, via right-click Breakpoint Properties.



Via the breakpoint properties you can

* Define a condition that restricts the activation of this breakpoint.
  + For example specify that a breakpoint should only become active after it has reached 12 or more times via the Hit Count property.
* You can also create a conditional expression.
  + The execution of the program only stops at the breakpoint, if the condition evaluates to true.
* This mechanism can also be used for additional logging, as the code that specifies the condition is executed every time the program execution reaches that point.

A watch-point is a breakpoint set on a field. The debugger will stop whenever that field is read or changed.

In the properties of a watchpoint you can configure if the execution should stop during read access (Field Access) or during write access (Field Modification) or both.

Exception breakpoint

To define an exception breakpoint click on the Add Java Exception Breakpoint button icon in the Breakpoints view toolbar.

A method breakpoint is defined by double-clicking in the left margin of the editor next to the method header.

You can configure if you want to stop the program before entering or after leaving the method.

Breakpoints for loading classes

To set a class load breakpoint, right-click on a class in the Outline view and choose the Toggle Class Load Breakpoint option.

A class load breakpoint stops when the class is loaded.

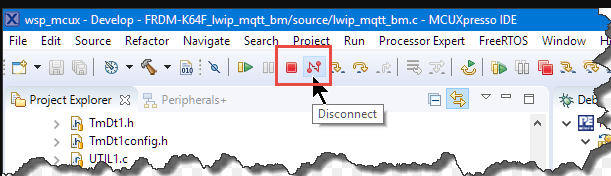
Drop to frame

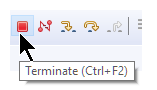
Eclipse allows you to select any level (frame) in the call stack during debugging and set the JVM to restart from that point.

This allows you to rerun a part of your program. Be aware that variables which have been modified by code that already run will remain modified.

To use this feature, select a level in your stack and press the Drop to Frame button in the toolbar of the Debug view.

<https://dzone.com/articles/mcuxpresso-ide-terminate-and-disconnect-a-debug-session>





While debugging with Eclipse, there are two different ways to end a debug session:

Terminate: stop the target under debug and disconnect from it and leave it in the stopped state.

Disconnect: leave the target as it is (running, suspended) and disconnect the debugger from the target.

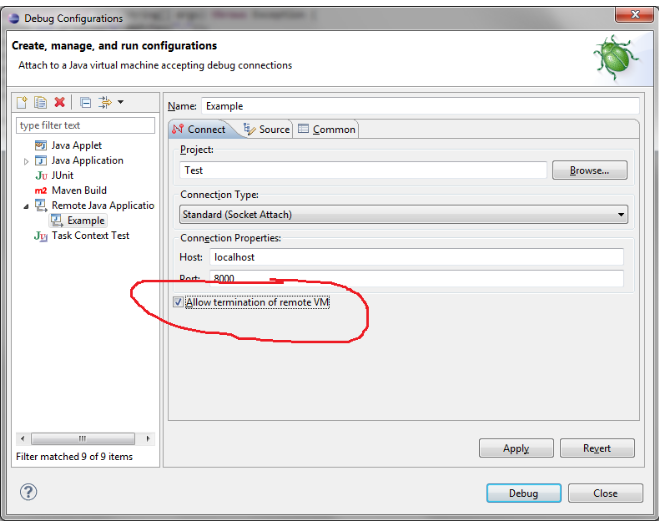
Terminate and Disconnect can mean different things for different Eclipse implementations.

https://stackoverflow.com/questions/19177716/how-to-terminate-a-remote-eclipse-debug-on-java

How to terminate a remote Eclipse debug on java

While remote debugging a java application I want to terminate the program. But the only buttons available are suspend and disconnect.

Suspend of course, does not terminate, and disconnect stops the debug but leaves the remote app in a whatever state where I then have to restart it.



Are you using Eclipse Juno ? I am experiencing same thing even if the termination of remote VM is enabled as suggested, in such a case you could switch to Debug perspective to kill the remove VM.