



# Integration for Eclipse via GDB

---

[TRACE32 Online Help](#)

[TRACE32 Directory](#)

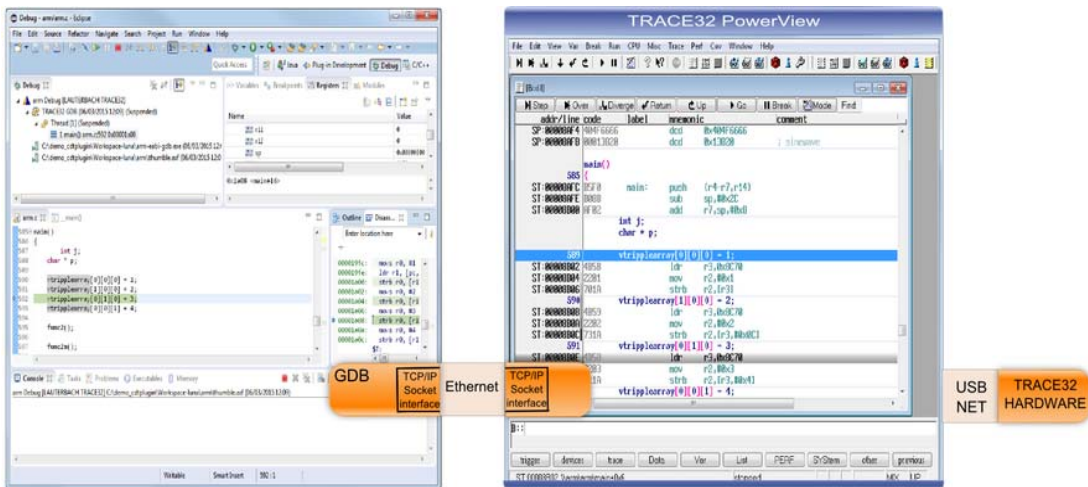
[TRACE32 Index](#)

<a href="#">TRACE32 Documents .....</a>	
<a href="#">3rd Party Tool Integrations .....</a>	
<a href="#">    <b>Integration for Eclipse via GDB .....</b></a>	<b>1</b>
<a href="#">    <b>Introduction .....</b></a>	<b>2</b>
<a href="#">        Supported EclipseIDE and CDT Version Combinations</a>	3
<a href="#">        Documentation Updates</a>	3
<a href="#">    <b>Installation .....</b></a>	<b>4</b>
<a href="#">        Requirements</a>	4
<a href="#">        Lauterbach TRACE32 Eclipse Plug-In</a>	5
<a href="#">            Install the Plug-In</a>	5
<a href="#">    <b>Create Launch Configurations .....</b></a>	<b>7</b>
<a href="#">    <b>Open TRACE32 from the Toolbar .....</b></a>	<b>13</b>
<a href="#">        Add the Lauterbach Logo Button</a>	13
<a href="#">    <b>Troubleshooting .....</b></a>	<b>14</b>
<a href="#">        Eclipse</a>	14
<a href="#">            Failed to Connect to TRACE32</a>	14
<a href="#">        Help Us Help You</a>	15
<a href="#">            Export the Eclipse Error Log</a>	15
<a href="#">            Export the Eclipse Configuration</a>	15

13-Apr-15 New document.

## Introduction

The TRACE32 PowerView can be used as GDB backend (see ["TRACE32 as GDB Back-End"](#) (backend\_gdb.pdf)). This GDB interface is useful to access the TRACE32 debug functionality (go, break, step, variable view, memory and register access...) from within Eclipse.



In order to allow you to use these features, Lauterbach offers an Eclipse plug-in with a simplified and adapted launch configuration.

**NOTE:** The Eclipse plug-in described in this document only simplifies the Eclipse configuration to control TRACE32 using the GDB interface and offers the possibility to configure and launch TRACE32 from within Eclipse. Controlling TRACE32 through Eclipse as GDB backend can also be done using the standard views offered by the Eclipse **Debug Configurations** window.

**NOTE:** The Eclipse plug-in described in this document is different from the Lauterbach Eclipse "loose coupling" plug-in which is described in ["Coupling for Eclipse"](#) (int\_eclipse.pdf).

## Supported Eclipse IDE and CDT Version Combinations

---

Eclipse 4.5 (Mars)	with CDT 8.7 & 8.8
Eclipse 4.4 (Luna)	with CDT 8.4 & 8.5 & 8.6
Eclipse 4.3 (Kepler)	with CDT 8.2 & 8.3
Eclipse 4.2 (Juno)	with CDT 8.1
Eclipse 3.7 (Indigo)	with CDT 8.0
Eclipse 3.6 (Helios)	with CDT 7.0
Eclipse 3.5 (Galileo)	with CDT 6.0

Other versions may work, but they have not been tested by Lauterbach.

Please note that **we cannot test, debug and support installations with modified Eclipse or CDT components.**

<b>NOTE:</b> For using the GDB connection, the CDT suitable version must be installed.
--

## Documentation Updates

---

The latest version of this document is available for download from:

[http://www.lauterbach.com/eclipse/cdtplugin/doc/int\\_eclipse\\_gdb.pdf](http://www.lauterbach.com/eclipse/cdtplugin/doc/int_eclipse_gdb.pdf)

## Requirements

---

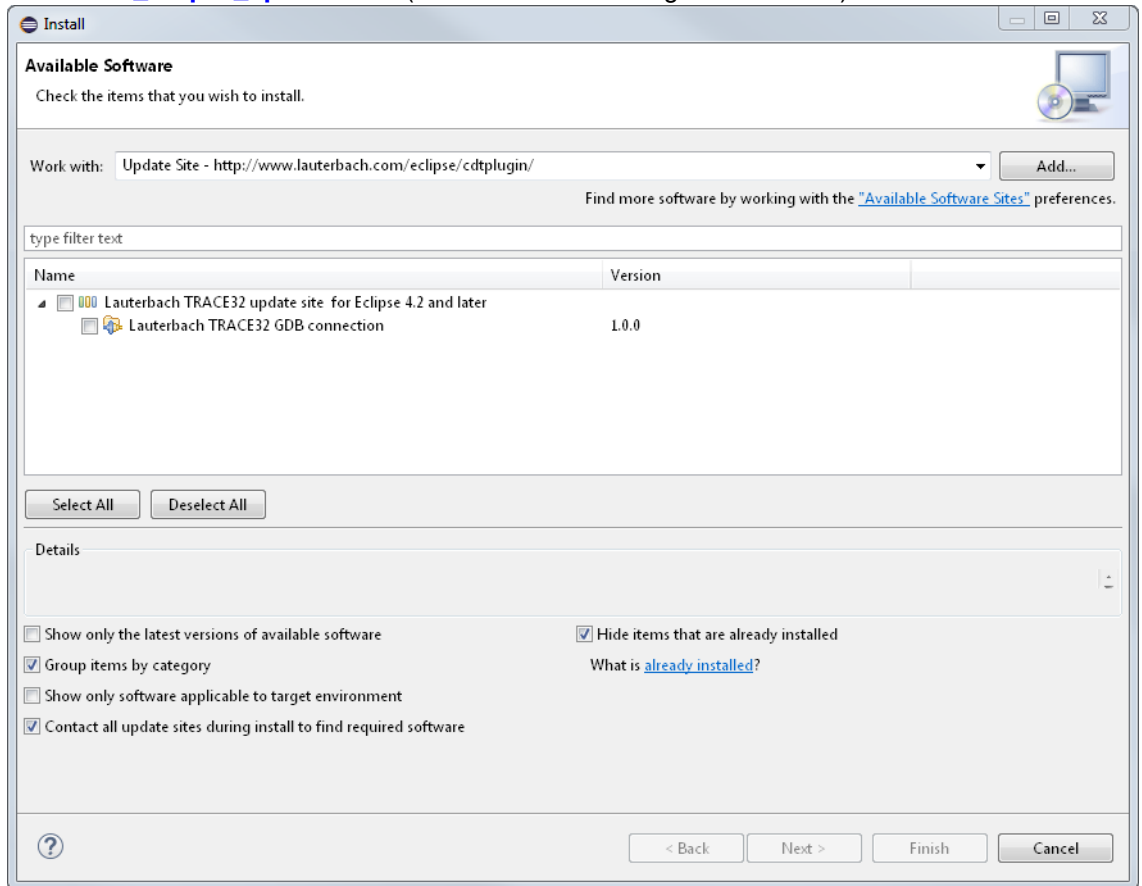
- An Eclipse installation, with the adequate “Eclipse C/C++ Development Tooling” CDT version as listed in the [Introduction](#) section, is required and it should be configured for Java Runtime Environment (**required minimum is JRE 1.6**).
- A TRACE32 installation of the required target architecture is needed. The minimum software version depends on the architecture used.
- In case of need, a correct **http-proxy configuration** for download and installation has to be configured.

**NOTE:** Set the **http-proxy configuration** in the Eclipse dialog **Window > Preferences**, in the section **Install/Update** or in **General > Network Connections**.

(The exact location depends on the Eclipse version used.)

## Install the Plug-In

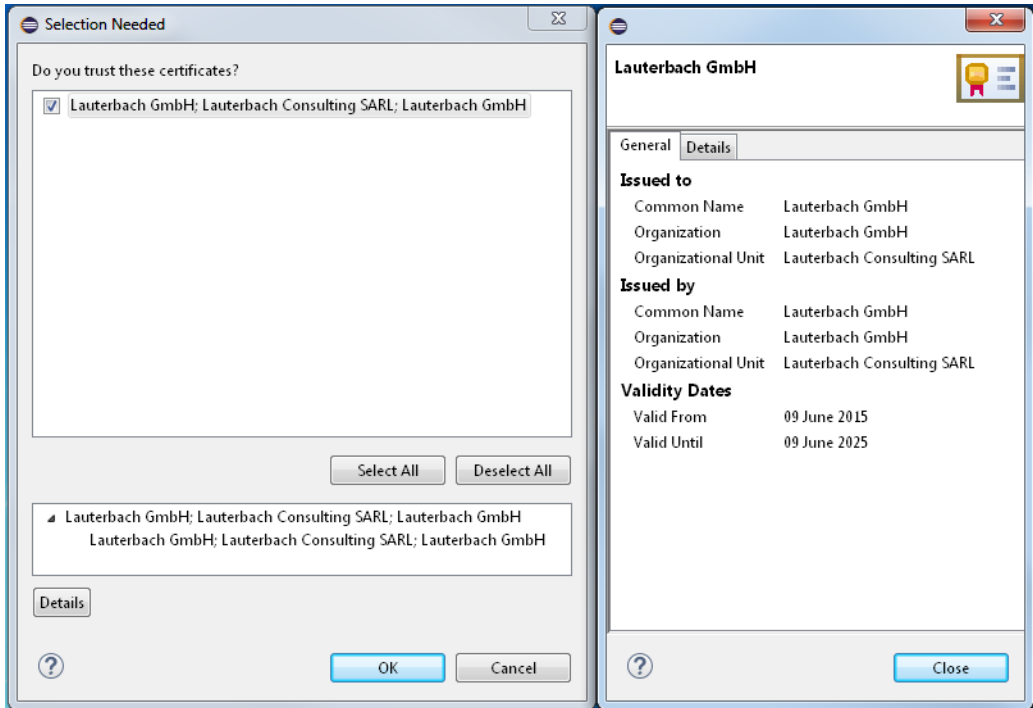
1. Open the Eclipse installation dialog **Help > Install New Software** (Eclipse 3.5 and later). The exact dialog sequence and naming has changed with almost every major Eclipse IDE version.
2. Enter the adequate path as shown in the screenshot below:
  - For Eclipse 4.2 and later: <http://www.lauterbach.com/eclipse/cdtplugin/>.
  - For Eclipse 3.7: [http://www.lauterbach.com/eclipse/cdtplugin/3.7\\_eclipse\\_updtasite/](http://www.lauterbach.com/eclipse/cdtplugin/3.7_eclipse_updtasite/).
  - For Eclipse 3.6 and 3.5: [http://www.lauterbach.com/eclipse/cdtplugin/3.6&3.5\\_eclipse\\_updatesite/](http://www.lauterbach.com/eclipse/cdtplugin/3.6&3.5_eclipse_updatesite/). (This version is no longer maintained)



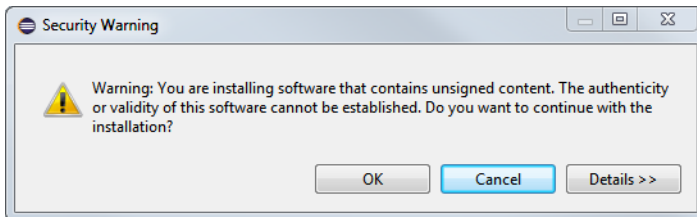
3. After selecting the adequate plug-in version in the **Install** dialog, please, follow the Eclipse installation wizard.

**NOTE:** Please make sure you have set up correct **http-proxy settings** for Eclipse before you start the plug-in installation.

4. For Eclipse 4.2 and later the plug-in is signed, so you may be prompted to confirm trust to the Lauterbach signature:



5. For Eclipse 3.7 and earlier, please confirm unsigned installation:



6. When prompted to do so, please restart Eclipse.

**Congratulations, plug-in installation is complete.**

# Create Launch Configurations

The Eclipse plug-in can start the TRACE32 PowerView via an Eclipse Launch Configuration. This document refers to Debug Configuration dialogs only.

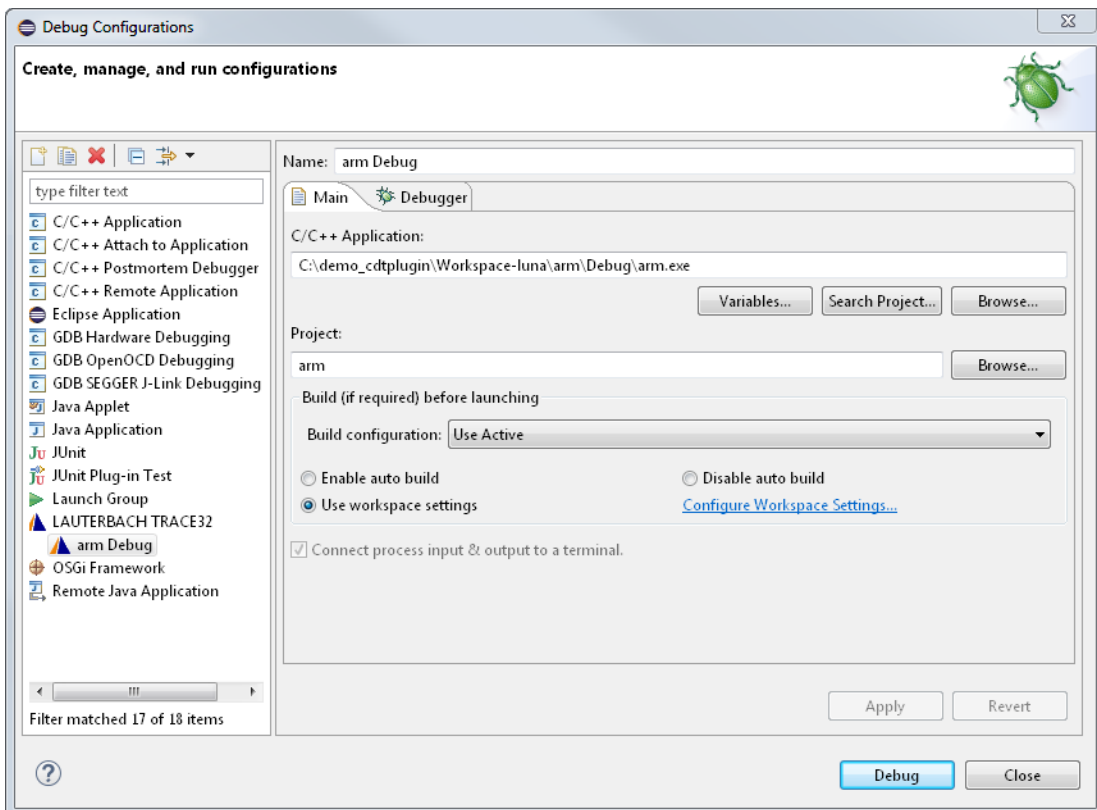
A demo exist under [http://www.lauterbach.com/eclipse/cdtplugin/demo/demo\\_cdtplugin.zip](http://www.lauterbach.com/eclipse/cdtplugin/demo/demo_cdtplugin.zip). It contains:

- An Eclipse (Luna) with the plugin installed.
- A TRACE32 Simulator for ARM.
- An Eclipse workspace that contains: an ARM project and a GDB executable for ARM (windows).

This demo was used to make the below screen shots. To be able to use it without any further changes, you need to unzip it under "C:\demo\_cdtplugin". If this is not possible, you will need to adapt the paths.

These steps are valid for Eclipse 4.4 (Luna), 4.3 (Kepler), 4.2 (Juno), 3.7 (Indigo), 3.6(Helios) and 3.5 (Galileo). The GDB connection is configured via an Eclipse **Launch Configuration**.

1. Open **Run > Debug Configurations** to set up a **Debug Launch Configuration**.
2. In the **Debug Configurations** dialog select **LAUTERBACH TRACE32** and add a new configuration with the context menu (right mouse button), opening this dialog:

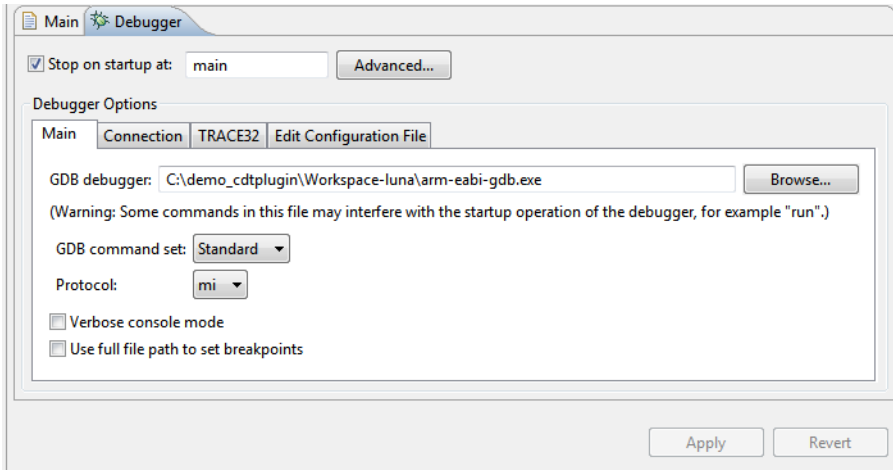


3. Choose and enter a name for your Debug Configuration.

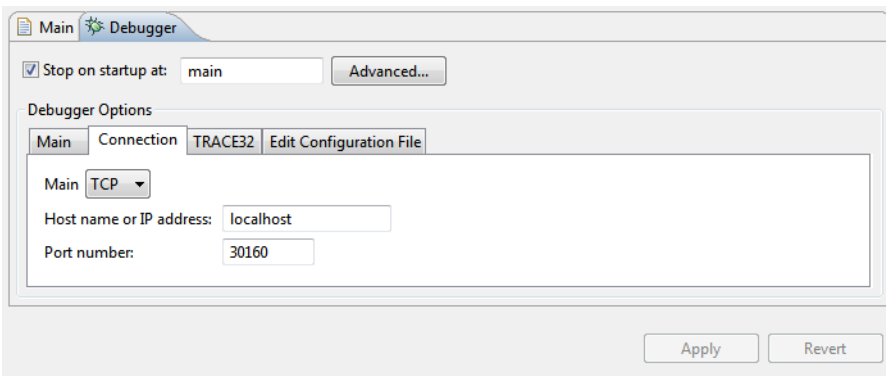
4. Set the **Project** field to the name of the Eclipse project with your source files.
5. In the field **C/C++ Application** enter the path to the application file.



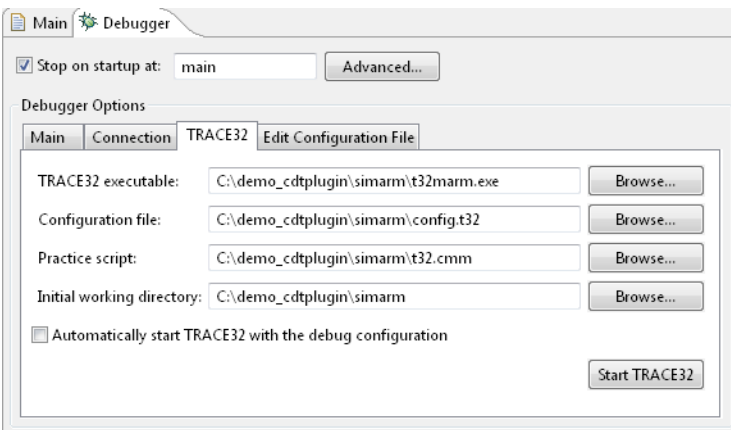
6. Go to the **Debugger** menu. Then, in the field **GDB debugger** enter the path to the GDB application.



7. Go to the connection list and enter your GDB configuration.



8. Go to the **TRACE32** tab. In the field **TRACE32 executable**, enter the path to the TRACE32 PowerView application. The file name of the TRACE32 executable depends on the target architecture (For example t32marm.exe for ARM).



**NOTE:** The TRACE32 GDB backend supports the architectures ARM, ARM64, PowerPC, MIPS32, SH, MicroBlaze, 68K, Intel x86, Intel x86 x64. Other architectures could be supported on request.

9. In the field **Configuration File** enter the TRACE32 configuration file to be used with the executable. TRACE32 has to be configured for use with the GDB backend by adding the following lines to the configuration file:

```
;-TRACE32 GDB backend configuration      <- mandatory empty line
GDB=NETASSIST                          <- optional comment line
PACKLEN=1024
PORT=30000                               ;PORT NUMBER
PROTOCOL=TCP
                                           <- mandatory empty line
```

**NOTE:**

PACKLEN specifies the maximum package length in bytes for the socket communication.

PORT specifies the port number for communication. The default port is 30000. If this port is already in use, try one higher than 30000.

PROTOCOL specifies the used protocol (TCP or UDP). Default is UDP.

An empty line before and after the text block is required!

Comment lines in the configuration file are ignored.

If the GDB configuration is in the end of the file and is correct, the launch configuration may show an error message. Try to add another empty line to the end of the file.

For more information, please refer to "[TRACE32 as GDB Back-End](#)" (backend\_gdb.pdf).

10. In the field **PRACTICE script**, you can enter the TRACE32 cmm file.

**NOTE:**

When started, If the PRACTICE script is not specified, TRACE32 will look for a file with the name **t32.cmm** in its initial working directory. If the file is found, the built-in PRACTICE interpreter will automatically read and execute this file as its **Startup Script**.

11. Set the **Initial Working Directory** for TRACE32 when it is launched by the plug-in. This directory will be the initial working directory for scripts started in TRACE32.

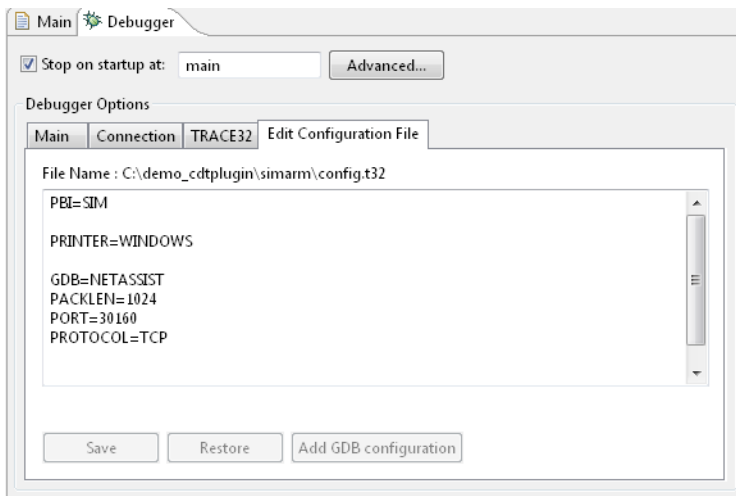
**NOTE:**

TRACE32 displays the initial working directory when you enter the **PWD** (=“print working directory”) command directly after TRACE32 starts.

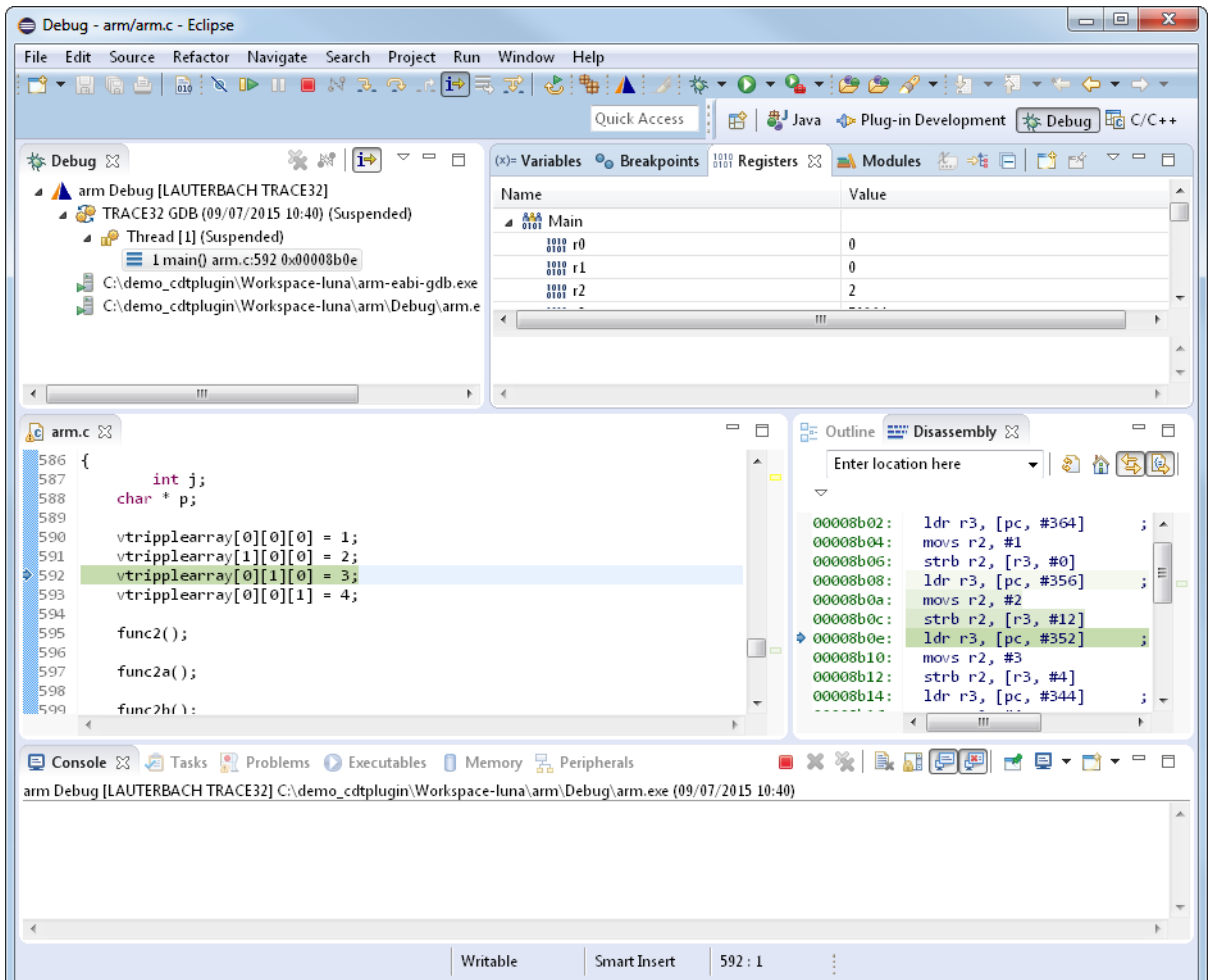
**NOTE:** TRACE32 can be opened with the **Start TRACE32** button or with pushing the **Debug** button. If the PRACTICE script needs too long time to finish, Eclipse may not be able to connect (time out). Please launch TRACE32 before starting debugging.

12. The **Edit Configuration File** tab gives the possibility to modify, save, restore and add automatically a GDB configuration. With this option the plug-in will use the port number value entered in the Connection tab.

**NOTE:** The TCP port is automatically corrected (any change in the **Connection** tab will be set to the configuration file and vice versa). This will only work if there is a valid GDB configuration bloc in the configuration file.



13. With a correct configuration in place, click the **Debug** button to launch TRACE32.



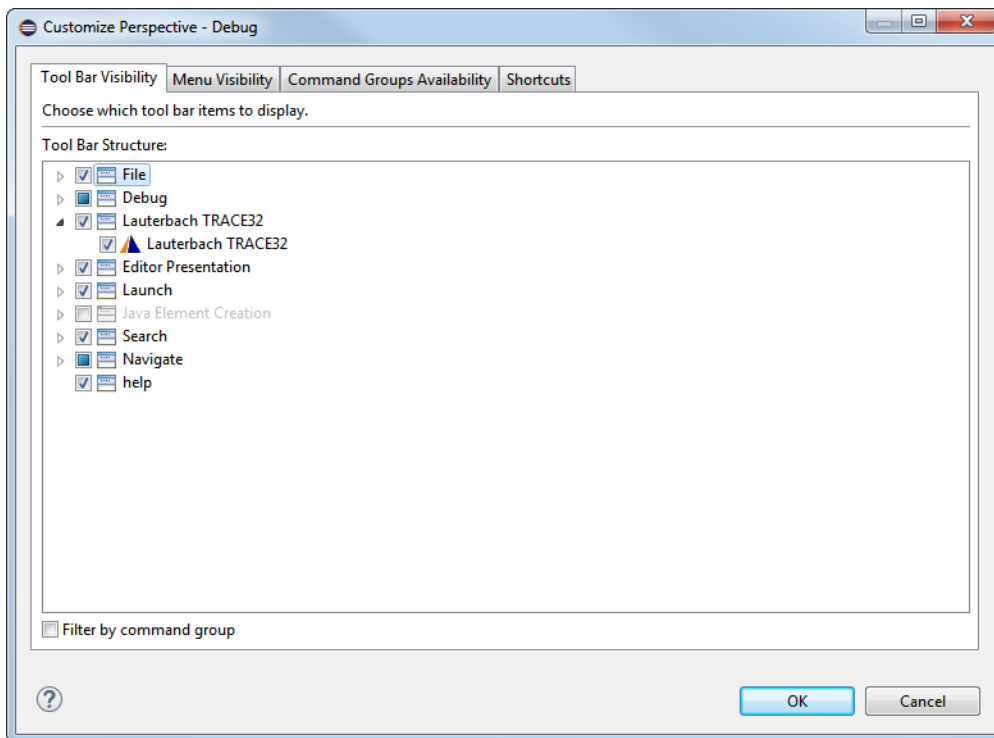
# Open TRACE32 from the Toolbar

The TRACE32 PowerView can also be executed through the Lauterbach Logo button in the Eclipse **C/C++ Perspective** toolbar. The **Lauterbach Logo** button executes the TRACE32 PowerView with parameters entered in the **Debug Configuration**.

## Add the Lauterbach Logo Button

Normally, the Lauterbach Logo button in the toolbar is automatically added. But, if it is not showed in the toolbar:

1. Open the **Windows** menu.
2. Choose **Customize Perspective**.
3. Activate the command group **Lauterbach TRACE32**.



## Eclipse

---

### Failed to Connect to TRACE32

---

Symptoms: Launching TRACE32 from Eclipse works, it starts up fine. However, after some time-out, Eclipse displays the message “Target selection failed”.

Please make sure the TRACE32 GDB configuration is set properly.

The TRACE32 configuration file must contains:

```
;-T32 API Access          <- mandatory empty line
GDB=NETASSIST            <- optional comment line
PACKLEN=1024
PORT=30000                ;PORT NUMBER
PROTOCOL=TCP              <- mandatory empty line
```

and the PORT number is the same as in the connection tab.

For a detailed example, please see [“Create Launch Configurations”](#), page 7.

## Export the Eclipse Error Log

---

Please include the full Eclipse **Error Log** as a file in your support request:

1. Open the **Error Log** view in Eclipse with **Window > Show View > Error Log**.
2. Click the **Export** icon on the **Error Log** view toolbar.
3. **Save** the log as a file.
4. Attach this file to your support request.

## Export the Eclipse Configuration

---

Export the Eclipse configuration settings in text form to the clipboard. With this we can check your Eclipse configuration for any missing or outdated components.

1. Open **Help > About Eclipse**.
2. Click the button **Installation Details**.
3. Select the **Configuration** tab.
4. Click **Copy to Clipboard**.
5. Paste the clipboard content into your support mail to Lauterbach.