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

Version 04-Apr-2016

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.

C Debug - anvienz - Edipte		0.0		TRACE32 PowerView
Fie Life Source Relator Navigate Search Proper Ran Window He Structure Control of the Structure	0-0-4-889-	()+()++ <b>0+</b> +++		He Life Yew Vie Beek Nam (29) Mor: Taxie Perl Can Window Help
Q4	ckAcces 🛛 🖉 🖓 Ina 💠 Pug	in Dealopment (D Debug () C/C++		
By Delay 11         Try of 100	No Vicalita: *a finalgonti 127 Eq Here 22 cili 22 cili	een II al Malan " 0 D-4 E II d " Vater " Addimiter		Image         Company         Company <thcompany< th=""> <thcompany< th=""> <thcom< td=""></thcom<></thcompany<></thcompany<>
2 ( ) () () () () () () () () () () () ()		Codies         Dom		S1         absolute         but         sy, the dt           S1         absolute         fat         f;           id         id         f;         id         f;           id         id         id         f;         id           id         id         id         f;         id           id         id         id         f;         id           id         id         id         id         id         id           id         id         id         id         id         id           id         id         id         id         id         id           id         id         id         id         <
Connect 12 (2 has 12 holows () honores on Shog (MCHRICHTACE) (2 honor, physical link part backweller i 1	X 2      white of (5673/3311201)	GDB TCP/IP Socket interface	Ethernet	TCPARe         Inv         Inv<

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 <b>Debug Configurations</b> window.
	the standard views offered by the Eclipse <b>Debug Configurations</b> 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.

**NOTE:** 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

©1989-2016 Lauterbach GmbH

### **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,<br/>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\_updtaesite/.
  - For Eclipse 3.6 and 3.5: http://www.lauterbach.com/eclipse/cdtplugin/
     3.6&3.5\_eclipse\_updatesite/. (This version is no longer maintained)

Install	
Available Software Check the items that you wish to install.	
Work with: Update Site - http://www.lauterbach.com/eclipse/cdtplugin/	Add  Find more software by working with the <u>"Available Software Sites"</u> preferences.
type filter text	
Name	Version
IIII Lauterbach TRACE32 update site for Eclipse 4.2 and later IIII Lauterbach TRACE32 GDB connection	1.0.0
Select All Deselect All	
Details	121
Show only the latest versions of available software	Hide items that are already installed
Image: Weight of the second secon	What is <u>already installed</u> ?
Show only software applicable to target environment	
U Contact an update sites during install to find required software	
?	< Back Next > Finish Cancel

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:

Selection Needed	22	
Do you trust these certificates?		Lauterbach GmbH
		General     Details       Issued to     Common Name       Organization     Lauterbach GmbH       Organizational Unit     Lauterbach Consulting SARL       Issued by     Common Name       Common Name     Lauterbach GmbH       Organization     Lauterbach GmbH
		Organizational Unit Lauterbach Consulting SARL Validity Dates Valid From 09 June 2015 Valid Until 09 June 2025
Select All Deselect , Lauterbach GmbH; Lauterbach Consulting SARL; Lauterbach GmbH Lauterbach GmbH; Lauterbach Consulting SARL; Lauterbach GmbH Details	H	
? OK Cancel		? Close

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

Security Warning	
Warning: You are install or validity of this softwa installation?	ng software that contains unsigned content. The authenticity re cannot be established. Do you want to continue with the
	OK Cancel Details >>

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

Congratulations, plug-in installation is complete.

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. 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:

Debug Configurations					23
Create, manage, and run config	urations			Ŕ	Š.
Image: Second Secon	Name: arm Debug Main Debugge C/C++ Application: C:\demo_cdtplugin\Wo Project: arm Build (if required) befor Build configuration: © Enable auto build © Use workspace settin V Connect process input	er] rkspace-luna\arm\Debug\arm re launching Jse Active ngs t & output to a terminal.	n.exe Variables Searc Disable auto build <u>Configure Workspace Setti</u>	th Project) Browse. Browse.	
Filter matched 17 of 18 items				Apply Revert	
?			E	Debug Clos	e

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.

Main Stop of	🌣 Debugger n startup at:	main	Advanced				
Debugge	r Options						
Main	Connection	TRACE32	Edit Configuration File				
GDB de	ebugger: C:\	demo_cdtpl	ugin\Workspace-luna\a	m-eabi-gdb.exe			Browse
(Warni	ng: Some con	nmands in t	his file may interfere with	the startup operation	ation of the deb	ugger, for exa	mple "run".)
GDB o	command set:	Standard	•				
Proto	col:	mi 🔻					
🗖 Ver	bose console i	mode					
🔲 Use	full file path t	o set breakp	oints				
						Apply	Revert

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

ebugger Options		
Main Connection TR/	CE32 Edit Configuration File	
Main TCP 🔻		
Host name or IP address:	localhost	
Port number:	30160	

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).

lain Connection TR	ACE32 Edit Configuration File	
TRACE32 executable:	C:\demo_cdtplugin\simarm\t32marm.exe	Browse
Configuration file:	C:\demo_cdtplugin\simarm\config.t32	Browse
Practice script:	C:\demo_cdtplugin\simarm\t32.cmm	Browse
initial working directory:	C:\demo_cdtplugin\simarm	Browse

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.

#### ©1989-2016 Lauterbach GmbH

9

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:

```
<- mandatory empty line
;TRACE32 GDB backend configuration
GDB=NETASSIST
PACKLEN=1024
PORT=30000
PROTOCOL=TCP
<- mandatory empty line</pre>
```

**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 looks 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 vise versa). This will only work if there is a valid GDB configuration bloc in the configuration file.

Main	Connection TR/	ACE32 Edit Configuration File	
File N	ame : C:\demo_cdt	olugin\simarm\config.t32	
PBI=	SIM		
PRIN	ITER=WINDOWS		
GDB	=NETASSIST		E
PAC	KLEN=1024 T=30160		
PRO	TOCOL=TCP		
			-

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

Debug - arm/arm.c - Eclipse					
File Edit Source Refactor Navigate Search Project Run	Window Help				
📑 🛨 🗄 🖕 🔜 🔪 🕨 II 🔳 🕺 🌫 과 🔝	5 🕫 🕹 🛸 👗	🌛 💠 • 🜔 • 9	🖌 🕶 😂 🔗 🕶 🖢 🕶	3 • ♥ ↔ • → •	
	Quick	Access 🕴 😰 🛛	lava 🛭 🚸 Plug-in Development	🏇 Debug 📴 C/C++	
🎋 Debug 🛛 🧏 🦓 📴 🗖 🗖	(x)= Variables 🔍 Bre	akpoints 1919 Registers	s 🛿 🛋 Modules 🛛 📩 🕫 🛛		
arm Debug [LAUTERBACH TRACE32]	Name		Value	*	
TRACE32 GDB (09/07/2015 10:40) (Suspended)	⊿ 👬 Main				
Inread [1] (Suspended)	1010 r0		0		
C:\demo_cdtplugin\Workspace-luna\arm-eabi-gdb.exe	1010 r1		0		
C:\demo_cdtplugin\Workspace-luna\arm\Debug\arm.e	0101 r2				
	•			•	
				*	
	4				
	C				
🔊 arm.c 🔀			📑 Outline 🔛 Disassembly	3	
586 {		×	Enter location here	- 🔊 🏠 🔄 🖳	
587 int ]; 588 char * p:			$\bigtriangledown$		
589			00008b02: 1dr r3, [	oc, #364] ; 🔺	
590 vtripplearray $[0][0][0] = 1;$			00008b04: movs r2,	¢1	
<pre>&gt;591 Vtripplearray[1][0][0] = 2; &gt;592 vtripplearray[0][1][0] = 3;</pre>			00008b06: strb r2,	r3, #0]	
593 vtripplearray[0][0][1] = 4;			00008b0a: movs r2, 4	£2 , #350 j	
594 595 func2():			00008b0c: strb r2,	r3, #12]	
596			● 00008b0e: 1dr r3,[] 00008b10: movs r2, 3	oc, #352] ; #3	
597 func2a();			00008b12: strb r2,	[r3, #4]	
598 599 func2h():		-	00008b14: ldr r3, [	oc, #344] ; 👻	
4		- F	< III	•	
🗉 Console 💥 🧔 Tasks 🐰 Problems 🜔 Executables 👖 Me	mory 🗟 Peripherals		× 🍇 🖹 🖬 🖅 🖓 🛛	* 🗉 + 🗂 + 😑 🕞	
				*	
4					
14.5		incort 502 1			
Wm	able smart.	nsert 592:1			

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 tollbar 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.

Customize Perspective - Debug	
Tool Bar Visibility Menu Visibility Command Groups Availability Shortcuts	
Choose which tool bar items to display.	
Tool Bar Structure:	
⊳ 🖉 🧮 File	
▷ 🔳 🔤 Debug	
a 👿 🔤 Lauterbach TRACE32	
🖉 🛕 Lauterbach TRACE32	
Image: Second	
Caunch     Constitution	
s V Search	
Navigate	
🕼 🔤 help	
(?)	OK Cancel

### **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:

	<- mandatory empty line
;T32 API Access	;<- optional comment line
GDB=NETASSIST	
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.

## Help Us Help You

#### **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.