

TRACE32 as TCF Agent

Release 02.2024



| TRACE32 Online Help | |
|---|----|
| TRACE32 Directory | |
| TRACE32 Index | |
| TRACE32 Documents | |
| TRACE32 as TCF Agent | 1 |
| Introduction | 4 |
| Restrictions | 5 |
| Documentation Updates | 6 |
| Related Documents and Tutorials | 6 |
| Initial Setup and Requirements | 7 |
| TRACE32 | 7 |
| Eclipse | 8 |
| Wind River Workbench | 10 |
| Synopsys MetaWare IDE | 12 |
| TRACE32 Setup | 15 |
| Installing the TRACE32 TCF Eclipse Plug-In | 15 |
| Option A: Manual Configuration | 17 |
| Option B: Select Executable and Configuration File | 22 |
| Configuration File | 24 |
| T32Start | 24 |
| Establish a Debug Session | 25 |
| Start TRACE32 | 25 |
| TCF Discovery | 27 |
| Manual Debug Target Setup | 29 |
| Open Debug Perspective Automatically | 31 |
| TRACE32 View | 32 |
| Troubleshooting | 33 |
| TRACE32 | 33 |
| TCF=(illegal command) | 33 |
| Eclipse | 33 |
| No TRACE32 PowerView instance under "Available Targets" | 33 |
| Cannot locate peer TCP: <ip>:<port></port></ip> | 33 |
| Help Us to Help You | 34 |
| Export the TRACE32 System Information | 34 |
| Export the Eclipse Error Log | 34 |
| Export the Eclipse Configuration | 35 |

| TCF Commands | | 36 |
|------------------------------|------------------------------|----|
| SYStem.TCFconfig | TCF-specific setups | 36 |
| SYStem.TCFconfig.TASKCONTEXT | Enable/disable task contexts | 36 |

Introduction

The Target Communication Framework (TCF) is a vendor-neutral lightweight network protocol mainly for communication with embedded systems (http://wiki.eclipse.org/TCF).

TRACE32 PowerView can be configured as a TCF agent. This TCF interface is useful to access the debug functionality of TRACE32 from within an Eclipse-based interface. Simultaneous usage of TRACE32 PowerView and Eclipse is also possible.



The TRACE32 TCF integration supports the following debugging features:

- Run control (Go, Break, Step,...)
- Software and on-chip breakpoints
- Register view
- Expressions view
- Memory view
- Display of the source code and the disassembly (mixed mode) for the selected context
- Stack trace
- Display of the operating system tasks in the Eclipse Debug view.
- Function sample-based profiling
- Debug Symbol Browser for the Wind River Workbench
- Multi-core debug in AMP and SMP

The TRACE32 TCF integration also provides a synchronization between TRACE32 PowerView and Eclipse. For example, setting a breakpoint or executing a single step at the TRACE32 side will be reported to Eclipse and vice versa.

| NOTE | |
|-------|--|
| | |
| NOTE: | The TRACE32 TCF integration is available for all architectures and compilers supported by TRACE32. |

NOTE: The debug symbols have to be loaded in TRACE32 PowerView and do not need to be loaded in Eclipse.

Restrictions

This solution is not a full integration of TRACE32 in the described IDEs. Not all features of TRACE32 PowerView are available in Eclipse or Wind River® Workbench. This solution should be only be used for debugging bare-metal applications or simple operating systems as FreeRTOS.

The following features are for instance not supported by this solution:

- Peripheral views, MMU and Cache views
- Memory access classes. The memory view in Eclipse only shows the current context.
- Trace
- MMU and Cache views
- OS Awareness
- Hypervisor debugging
- FLASH programming

Please contact support@lauterbach.com for more information.

Documentation Updates

The latest version of this document is available for download from: www.lauterbach.com/pdf/app_tcf_setup.pdf

Related Documents and Tutorials

- For information about how to install TRACE32, see "TRACE32 Installation Guide" (installation.pdf).
- For a video tutorial about TRACE32 as a TCF agent for Wind River Workbench, visit: support.lauterbach.com/kb/articles/demo-of-the-trace32-integration-to-the-wind-riverworkbench
- For a video tutorial about TRACE32 as a TCF agent for Eclipse, visit: support.lauterbach.com/kb/articles/demo-of-the-trace32-integration-to-eclipse

In this chapter:

- TRACE32
- Eclipse: If you want to integrate TRACE32 with Eclipse, then skip the *Wind River Workbench* section in this document.
- Wind River Workbench:
 - If you want to integrate TRACE32 with Wind River Workbench, then you have to take the additional steps described in section "Wind River Workbench".
 - After that, continue with section "TRACE32 Setup", page 15
- Synopsys MetaWare IDE: If you want to integrate TRACE32 with the MetaWare IDE (a special version of Eclipse)

TRACE32

For information about how to install TRACE32 under MS Windows, see "MS Windows" in TRACE32 Installation Guide, page 21 (installation.pdf). We recommend that you install TRACE32 on the system path suggested by the installer: C:\T32.

For information about how to install TRACE32 under Linux, see "**PC_LINUX**" in TRACE32 Installation Guide, page 23 (installation.pdf).

A TRACE32 version from **February 2016 or later** is required. If the TRACE32 version is too old, then you will get an error box with the message "**TCF=(illegal command**")" when trying to start TRACE32 PowerView as a TCF agent.

• To check the TRACE32 version, choose **Help** menu > **About TRACE32**.

For a description of how to configure and start TRACE32 as a TCF agent, see "TRACE32 Setup", page 15.

Eclipse

According to the Eclipse TCF documentation, the following components are required:

- JDK 1.8.0 or later
- Eclipse SDK 3.8 or later
- CDT (C/C++ Development Tools) SDK 8.1 or later

For more information about the minimal required setup, please refer to the TCF Wiki page (http://wiki.eclipse.org/TCF).

| NOTE: | For Eclipse 3.7 (Indigo) and 3.6 (Helios) you need to install the "Target |
|-------|---|
| | Communication Framework (Incubation)" plug-in from |
| | http://download.eclipse.org/tools/cdt/releases/indigo. |

To install the TCF C/C++ Debugger in Eclipse (3.8 or newer):

- 1. Choose Help menu > Install New Software.
- 2. From the **Work with** list, select -- **All Available Sites** -- or select an update site for your Eclipse version under http://www.eclipse.org/tcf/downloads.php.
- 3. Type "TCF" in the search field.

You may have to wait for the list to be populated.

4. Select the **TCF C/C++ Debugger**.

| C Install | | X |
|---|--|------------------|
| Available Software Check the items that you wish to install. | | |
| Work with:All Available Sites | Ac Find more software by working with the <u>"Available Software Sites"</u> pre- | ld eferences. |
| TCF | | R_ |
| Name | Version | ^ |
| W C/C++ Remote (over TCF/TE) Run/Debug Launcher W TCF Terminal (Console) View W Mobile and Device Development | 1.2.0.201405271222 1.2.0.201406042322 | |
| C/C++ Remote (over TCF/TE) Run/Debug Launcher C/C++ Debugger CF C/C++ Debugger | 1.2.0.201405271222 1.2.0.201405271222 | = |
| Arrow TCF Remote System Explorer add-in Arrow TCF Target Explorer | 1.2.0.201405271224 1.2.0.201405271222 | |
| Grant CF Terminal (Console) View | 1.2.0.201406042322 | - |
| Select All Deselect All I item selected | | |
| A full-featured C/C++ debugger for local or remote systems running a TCF a | gent.TCF is an extensible, lightweight protocol for universal target access. | ± More |
| Show only the latest versions of available software | Hide items that are already installed | |
| Group items by category | What is <u>already installed</u> ? | |
| Show only software applicable to target environment | | |
| Contact all update sites during install to find required software | | |
| ? | < Back Next > Finish Ca | ncel |

- 5. Click the various **Next** buttons and follow the instructions of the install wizard to finish the installation.
- 6. Restart Eclipse.

| NOTE: | For the Wind River Workbench 3.3, you need to install the "Target Communication Framework (Incubation)" plug-in from |
|-------|---|
| | http://download.eclipse.org/tools/cdt/releases/indigo. |
| | integration. |

You need to enable the TCF C/C++ Debugger in the Wind River Workbench.

- 1. Choose **Windows** menu > **Preferences**.
- 2. In the **Preferences** dialog, click **Wind River** > **Capabilities**.
- 3. Select the check box **TCF C/C++ Debugger**.

| W Preferences | |
|--|---|
| type filter text General C/C++ | Capabilities ← ▼ ⇒ ▼ ✓ Prompt when enabling capabilities Canabilities: |
| Help Install/Update Remote Development RSS/Atom Feed View Run/Debug Target Explorer Team Terminal Validation Wind River | |
| Build Console Build Properties Capabilities FTP/TFTP Server Host Tools Execution XML | Use the C/C++ debugger to debug systems running a TCF agent. Restore Defaults Apply OK Cancel |

Moreover, you need to disable the launch configuration filter for the Target Communication Framework.

- 1. Choose Windows menu > Preferences
- 2. In the **Preferences** dialog, click **Run/Debug** > **Launching** > **Launch Configurations**.
- 3. Clear the check box **Target Communication Framework**.



The MetaWare IDE is based on Eclipse. Thus you can basically you can use TCF the same way than described for native Eclipse before.

Please note that the MetaWare IDE contains proprietary changes to Eclipse. As a result Lauterbach can't ensure that controlling TRACE32 via TCF is fully functional. Thus it is recommended to use native Eclipse.

To install the TCF C/C++ Debugger in Synopsys MetaWare IDE:

- 1. Choose Help menu > About MetaWare IDE > Installation Details > Features
- 2. Check the version number of the Eclipse Platform. This is the Eclipse version on which you copy of MetaWare IDE is based on. You must have version 4.2 or higher.

| MetaWare | IDE | based on Eclipse Platform | Comment |
|---------------|------|---------------------------|-------------------------|
| L-2016.06 | 10.3 | Eclipse 4.4 (Luna) | TCF successfully tested |
| L-2016.03 | 10.2 | Eclipse 4.4 (Luna) | |
| K-2015.12 | 10.1 | Eclipse 4.4 (Luna) | |
| K-2015.09 | 10.0 | Eclipse 4.4 (Luna) | |
| J-2015.03 | 9.8 | Eclipse 4.4 (Luna) | |
| J-2014.12 | 9.7 | Eclipse 4.4 (Luna) | |
| J-2014.06-SP1 | 9.6 | Eclipse 4.4 (Luna) | |
| J-2014.06 | 9.5 | Eclipse 4.3 (Kepler | |
| I-2013.12.1 | 9.4 | Eclipse 4.3 (Kepler) | |
| I-2013.12 | 9.3 | Eclipse 4.3 (Kepler) | |

3. Choose Help menu > Install New Software...

4. In dialog windows "Install" add the in the Work with box the update site according to the version of the Eclipse platform you got before.

| Eclipse Platform | Update Site (p2 Repository) |
|------------------------|--|
| Eclipse 4.14 (2019-12) | http://download.eclipse.org/releases/2019-12 |
| Eclipse 4.13 (2019-09) | http://download.eclipse.org/releases/2019-09 |
| Eclipse 4.12 (2019-06) | http://download.eclipse.org/releases/2019-06 |
| Eclipse 4.11 (2019-03) | http://download.eclipse.org/releases/2019-03 |
| Eclipse 4.10 (2018-12) | http://download.eclipse.org/releases/2018-12 |
| Eclipse 4.9 (2018-09) | http://download.eclipse.org/releases/2018-09 |
| Eclipse 4.8 (Photon) | http://download.eclipse.org/releases/photon |
| Eclipse 4.7 (Oxygen) | http://download.eclipse.org/releases/oxygen |
| Eclipse 4.6 (Neon) | http://download.eclipse.org/releases/neon |
| Eclipse 4.5 (Mars) | http://download.eclipse.org/releases/mars |
| Eclipse 4.4 (Luna) | http://download.eclipse.org/releases/luna |
| Eclipse 4.3 (Kepler) | http://download.eclipse.org/releases/kepler |

You can get a full list of all "p2 Repositories" at https://wiki.eclipse.org/Simultaneous_Release

- 5. Type "TCF" in the search field. (You may have to wait for the list to be populated.)
- 6. Select the **TCF C/C++ Debugger**.
- 7. Click on **Next** and follow the dialogs, which guide you through the installation process.

It's recommended to install the TRACE32 TCF Eclipse Plug-In as described below.

The MetaWare IDE uses several special proprietary views (child windows) when it is connected to a MetaWare (or SeeCode) debugger. When using TCF you have to use the native Eclipse debug views. To get the right views used **Window** menu > **Show View** (> Other...)

| Data to display | MetaWare view | Native Eclipse View | TRACE32 Command |
|---------------------------|----------------------------|---------------------|---------------------|
| Core Register | Register (CDI) Register Re | | Register.view |
| Breakpoints | Breakpoints (MetaWare)* | Breakpoints | Break.List |
| Watchpoints | Watchpoints | Breakpoints | Break.List |
| Disassembled Code | Disassembly (MetaWare) | Disassembly | List.Mix |
| Raw Memory | Memory (MetaWare)* | Memory | Var.DUMP |
| Local Variables | Locals* Variables Va | | Var.Local |
| Global Variables | Global Variables* | Expressions | Var.Watch |
| HLL Expressions | Expressions (MetaWare)* | Expressions | Var.Watch |
| Call Stack | Call Stack* | Debug | Frame.view |
| ELF Sections | Modules (MetaWare) | Modules | sYmbol.List.SECtion |
| Auxiliary Register | Auxiliary register | N/A | Data.dump AUX:0 |
| Hierarchic Symbol Tree | Executable | | sYmbol.Browse |

(* If marked with '*' you can also use the native Eclipse view instead.)



Don't mix up the Target Communication Framework (TCF) with the ARC target configuration files (TCF files). Both have nothing in commen except theire abbreviation.

Installing the TRACE32 TCF Eclipse Plug-In

Lauterbach offers an Eclipse plug-in with a simplified and adapted launch configuration. Using this plug-in, you can configure and start TRACE32 from within Eclipse or the Wind River Workbench.

To install the TRACE32 TCF Eclipse plug-in:

- 1. Choose **Help** menu > **Install New Software**.
- 2. In the Work with box, type the update site: https://www.lauterbach.com/eclipse/tcf
- 3. Press Enter.
- 4. Make the selection shown below.

| € Install | | X |
|--|---------------------------------------|------------|
| Available Software Check the items that you wish to install. | | |
| Work with: http://www.lauterbach.com/eclipse/tcf | • | Add Manage |
| type filter text | | |
| Name | Version | |
| ♥ IIII Lauterbach TRACE32 TCF connection ♥ ♣ Lauterbach TRACE32 TCF connection | 1.2.7 | |
| | | |
| | | |
| Select All Deselect All Litem selected | | |
| Details | | |
| | | Ĵ. |
| Show only the latest versions of available software | Hide items that are already installed | |
| 🕼 Group items by category | What is <u>already installed</u> ? | |
| Show only software applicable to target environment | | |
| Contact all update sites during install to find required software | | |
| | | |
| | | |

- 5. Click the various **Next** buttons and follow the instructions of the install wizard to finish the installation.
- 6. Restart Eclipse.
- 7. Choose **Run** menu > **Debug Configurations**.

You should now have the **TRACE32 TCF** configuration in the **Debug Configurations** window, as shown below.

| Debug Configurations | | × |
|--|---|------------|
| Create, manage, and run configura Connect to Lauterbach TRACE32 over th | tions he Target Communication Framework | to. |
| Ype filter text C C/C++ Application C C/C++ Application C C/C++ Postmortem Debugger C C/C++ Remote Application Launch Group Launch Group Launch Group Launch Group TRACE32 TCF TRACE32 TCF | Configure launch settings from this dialog: Press the 'New' button to create a configuration of the selected type. Press the 'Duplicate' button to copy the selected configuration. Press the 'Delete' button to remove the selected configuration. Press the 'Delete' button to configure filtering options. Press the 'Filter' button to configure filtering options. Edit or view an existing configuration by selecting it. | |
| Filter matched 8 of 8 items | De | ebug Close |

- 8. Double-click **TRACE32 TCF** to access the new **TRACE32** tab.
- 9. On the **TRACE32** tab, choose one of the following configuration methods:
 - Option AIf you have not worked with TRACE32 before, we recommend that you
configure the different TRACE32 settings in the Eclipse plug-in by selecting
the Manual configuration option.
See "Option A: Manual Configuration", page 17.
 - Option BIf you already have a working TRACE32 configuration, we recommend that
you specify the TRACE32 executable and the configuration file directly by
selecting the Select executable and configuration file option.
See "Option B: Select Executable and Configuration File", page 22.

Option A: Manual Configuration

1. If you want to configure the TRACE32 options manually, select **Manual configuration** from the **TRACE32 configuration** drop-down list in Eclipse.

The fields for a manual configuration are will be displayed on the TRACE32 tab.

- 2. Make your settings.
 - For a description of the fields on the TRACE32 tab, see tables below.
 - A temporary configuration file will be created in the default temporary directory when TRACE32 is started.
 - For examples of the connection types USB and Ethernet, click here.

| Debug Configurations | X |
|--|---|
| Create, manage, and run configurations | |
| Connect to Lauterbach TRACE32 over the Target Communication Framework | |
| □ □ ■ ⇒ ▼ Name: New_configuration | |
| type filter text 🗰 Target 🗞 Download 🚡 Symbol Files 🔮 Path Ma | p 🙀 Source 🔲 Common 🕢 TRACE32 |
| C/C++ Application C/C++ Attach to Applic. TRACE32 configuratic :: Manual configuration | Start TRACE32 |
| C C/C++ Postmort A Parameters to launch 1000002 | |
| Launch Group Device: Debugger | ▼ |
| Target Communication Connection Type: USB | • |
| TRACE32 TCF Node name / IP Address: | |
| Architecture: ARM | • |
| B Paths: | |
| Working path: | Browse |
| System path: | Browse |
| C Interfaces: | |
| TCF Discovery Port value: 1534 | |
| TRACE32 API Port value: 20000 | |
| Display: | |
| Title: TRACI | 32 PowerView For ARM |
| Run TRACE32 PowerView in background | |
| E -Start-up Script: | |
| Source: File | |
| | |
| File: | Browse |
| Parameters: | |
| Automatically start TRACE32 with the debug configur Target ID: Delay starting TCF (s): | ation (TCF port number must be specified) |
| | |
| Filter matched 9 of 9 items | Revert Apply |
| ? | Debug Close |

3. Click **Apply** when you are done.

You are now ready to start TRACE32 as described in "Start TRACE32", page 25.

| Device | Select if you are using a JTAG Debugger or a TRACE32 Instruction Set Simulator. |
|---------------------------|--|
| Connection Type | For a JTAG Debugger, select if you have a USB or Ethernet connection to the Lauterbach debugger hardware. |
| Node Name / IP Address | For USB connection, optionally set the name of the connected device. This option is useful if multiple Lauterbach debuggers are connected per USB to one PC. For Ethernet connection, set the IP address or the host name of the used Lauterbach debugger bardware. |
| | useu Lauterbach debugger hardware. |
| Architecture | From the drop-down list, select the used target architecture (e.g. Arm, TriCore). |

[B] Paths

| Working Path | Active directory after starting the TRACE32 instance. |
|--------------|---|
| System Path | Directory where the executable and system files of TRACE32 are located (e.g. C:\T32). |
| | If you have observed our recommendation and installed TRACE32 on the system path suggested by the installer, you can ignore the step-by-step procedure below. |

To determine the system path of a particular TRACE32 installation:

- 1. Start TRACE32.
- 2. Type at the TRACE32 command line:

```
PRINT OS.PresentSystemDirectory()
   ;the message line below the TRACE32 command line
   ;displays the system path of this particular
   ;TRACE32 installation
```

[C] Interfaces

| TCF Discovery | Enable/disable the TCF discovery. If the discovery is disabled, a TCF port number must be specified (default: 1534). The TCF discovery is a mechanism where agents advertise their peers by sending UDP packets to other agents. |
|---------------|---|
| TRACE32 API | Enable/disable the TRACE32 Remote API |

| Title | Set the window title of the TRACE32 instance. The title will be displayed in the Name column under Available Targets if the TCF discovery is enabled. |
|---|---|
| Run TRACE32 PowerView in background | Start TRACE32 PowerView without a graphical user interface. TRACE32 can be terminated using the Eclipse Terminate button. |

[E] Start-up Script

| Source | When a TRACE32 instance starts, the PRACTICE script autostart.cmm is executed, which then calls the following scripts: system-settings.cmm (from the TRACE32 system directory, usually C:\t32) user-settings.cmm (from the user settings directory: on Windows %APPDATA%\TRACE32 or ~/.trace32 otherwise) work-settings.cmm (from the current working directory) In the TRACE32 TCF plug-in you can specify an additional PRACTICE script which is automatically started afterwards. The TRACE32 TCF plug-in supports two types of start-up scripts: File Built-in Script When Source is set to File, the script assigned to the File item will be executed. |
|-----------------|--|
| File | If Source is set to File , specify the start-up script (*.cmm) here. |
| Parameters | Set the parameters that are passed to the start-up script (*.cmm) from File . |
| Built-in Script | The start-up script can be edited and stored directly. |

[F] Start TRACE32 with the debug configuration

| Target ID | This option can be used only if TCF discovery is disabled. A new target with a specific port must be declared first (see Manual Debug Target Setup). The new "Target ID" should be then written into this box. |
|-----------|---|
| Delay | An optional delay time until TRACE32 completes its start-up. |

NOTE: The TRACE32 TCF plug-in automatically creates a TRACE32 configuration file when you start TRACE32 from within Eclipse.

| onnext to Lutebach TRACE32 over the Target Communication Framework Image: | eate, manage, and run configurations | | - |
|---|---|--|-----------------------------------|
| Image: Second | onnect to Lauterbach TRACE32 over the Targe | Communication Framework |) (|
| er matched 10 of 10 items Image: Construction of the constru | | Name Manual config USB | |
| CVC++ Application Vorget Vo | pe filter text | Transet (Sp. Download (Ex. Symbol Files (R. Path Map (Ex. Source (E. Common (A. TRACE3)) | |
| Image: Control to Application Image: Control to Bugger Iman | C/C++ Application | | |
| Image: CVC++ Restmonted Debugger Image: CVC++ Restmonted Application Launch Group Launch Group Launch Group (Deprected) TRACE32 TCF Manual_config_USB Vorties Restmanted Debugger Connection Type: USB Manual_config_USB Path: Working path: CVS2 Browse System path: CVS2 System path: CVS2 Browse Interfaces: TCF Discovery Port value Display: Title: TRACE32 PowerView For ARM Run TRACE32 PowerView in background Start-up Script: Source File CVT32/demolarm/hardware/omap44xx/pandaboard/mini/demo_sizev_sar-ram.cmm Parameters: Automatically start TRACE32 with the debug configuration (TCF port number must be specified) Target ID: Delay starting TCF (s): Person Revent Appby | C/C++ Attach to Application | TRACE32 configuration: Manual configuration | Start TRACE32 |
| Lauch Group (Deprected) Target Communication Framework A Manual, config. USB | C/C++ Remote Application | Parameters to launch TRACE32 | |
| cannection Type USB Target Communication Framework TRACE32 TCF Manual_config_USB Node name / IP Address: Architecture: ARM Paths: Working path: C\132 Working path: C\132 System path: C\132 Browse System path: System path: C\132 Browse Browse Interfaces: TCF Discovery Port value: TAACE32 API Port value: 20000 Display: Title: Title: TRACE32 PowerView For ARM Run TRACE32 PowerView in background Start-up Script: Source: File File: C\132/demo\arm\hardware\iomap44x\pandaboard\minh\demo_sieve_sar-ram.cmm Parameters: Automatically start TRACE32 with the debug configuration (TCF port number must be specified) Target ID: Delay starting TCF (s): Delay starting TCF (s): Revert | Launch Group | Device: Debugger | • |
| A TRACE32 TCF Manual_config_Lthemet Manual_config_USB Node name / IP Address: Architecture: ARM Paths: Browse System path: c\ts2 System path: c\ts2 Interfaces: TCF Discovery Port value: ISSI TTACE32 API Port value: Display: Ttrle Ttrle TRACE32 PowerView For ARM Run TRACE32 PowerView in background Start-up Script: Source: File File: C\T32/demo\amn\hardware\omp44xo\pandaboard\min\hdemo_sieve_sar-ram.cmm Parameters: Automatically start TRACE32 with the debug configuration (TCF port number must be specified) Target ID: Delay starting TCF (s); Delay starting TCF (s); Revett | Target Communication Framework | Connection Type: USB | • |
| Architecture: ARM Paths: Working path: c/t32 Paths: Working path: c/t32 Browse System path: c/t32 Browse Interfaces: TCF Discovery Port value: 1534 TRACE32 API Port value: 20000 Display: Title: TRACE32 PowerView For ARM Run TRACE32 PowerView in background Start-up Script: Source: File File: C/t32\demo\arm\hardware\omap44a0\pandaboard\mini\demo_sieve_sar-ram.cmm Browse Edit Parameters: Delay starting TCF (s): Revert Apply | TRACE32 TCF | Node name / IP Address: | |
| er matched 10 of 10 items | Manual_config_USB | Architecture: ARM | • |
| er matched 10 of 10 items Working path: c:\t32 Browse System path: c:\t32 Browse System path: c:\t32 Browse | | Paths: | |
| er matched 10 of 10 items | | Working path: c:\t32 | Browse |
| er matched 10 of 10 items | | System path: c:\t32 | Browse |
| er matched 10 of 10 items | | Interfaces: | |
| er matched 10 of 10 items | | ▼ TCF Discovery Port value: 1534 | |
| er matched 10 of 10 items | | TRACE32 API Port value: 20000 | |
| er matched 10 of 10 items | | Display: | |
| er matched 10 of 10 items | | Title: TRACE32 PowerView For ARM | |
| er matched 10 of 10 items | | Run TRACE32 PowerView in background | |
| er matched 10 of 10 items | | Start-up Script: | |
| er matched 10 of 10 items | | Source: File | |
| er matched 10 of 10 items | | File: C:\T32\demo\arm\hardware\omap44xx\pandaboard\mini\demo_sieve_sar-ram.cmm | Browse Edit |
| er matched 10 of 10 items | | | |
| er matched 10 of 10 items | | Parameters: | |
| er matched 10 of 10 items | | Automatically start TRACE32 with the debug configuration (TCF port number must be specified) | |
| er matched 10 of 10 items | | Target ID: | |
| er matched 10 of 10 items Revert Apply | | Delay starting TCF (s): | |
| er matched 10 of 10 items Revert Apply | | | |
| er matched 10 of 10 items Revert Apply | | | |
| er matched 10 of 10 items | | | Apply |
| | er matched 10 of 10 items | | Арру |
| | | | Close |

| | 31 |
|---|---|
| ate, manage, and run configurations onnect to Lauterbach TRACE32 over the Targ | Jet Communication Framework |
| 🗎 🗶 🕒 🎲 🕶 | Name: Manual_config_Ethernet |
| pe filter text |] 🔤 Target 🥎 Download 📻 Symbol Files 🔒 Path Map 💱 Source 🔲 Common 🚹 TRACE32 |
| C/C++ Application C/C++ Attach to Application | TRACE32 configuration: Manual configuration |
| C/C++ Postmortem Debugger | Parameters to launch TRACE32 |
| Launch Group | Device: Debuaaer 🗸 |
| Launch Group (Deprecated) Target Communication Framework | Connection Type: Ethernet |
| TRACE32 TCF | Node name / IP Address: pod0.example.com |
| Manual_config_USB | Architecture: TriCore |
| | Paths: |
| | Working path: |
| | System path: |
| | Interfaces: |
| | V ICF Discovery Port value: 1534 |
| | Diroland |
| | Title: TRACE32 PowerView For TriCore |
| | Run TRACE32 PowerView in background |
| | Start-up Script: |
| | Source: File • |
| | File: C:\T32\demo\tricore\hardware\triboard-tc2x5\tc275t\tc275t\tc275t_demo.cmm Browse Edit |
| | Parameters: |
| | Automatically start TRACE32 with the debug configuration (TCF port number must be specified) |
| | Target ID: |
| | Delay starting TCF (s): |
| | |
| | |
| er matched 10 of 10 items | Revert Apply |
| | |

You can, instead of configuring the TRACE32 parameters manually, directly set the path to the TRACE32 executable and configuration file.

1. From the **TRACE32 configuration** drop-down list, **Select Executable and configuration file**.

The fields for a direct configuration are will be displayed on the TRACE32 tab.

2. Make your settings.

For a description of the fields on the **TRACE32** tab, see tables below.

| Debug Configurations | | | × |
|---|----------------------------------|---|---------------------------|
| Create, manage, and run configu | urations | | |
| Connect to Lauterbach TRACE32 ove | er the Target Communication Fram | ework | |
| | Name: New configuration | | |
| type filter text | 📟 Target 🥎 Download 🚋 Sy | ymbol Files 🔒 Path Map 🧤 Source 🔲 Common 加 TRACE32 | |
| C/C++ Application C/C++ Attach to Applicatic | TRACE32 configuration Select | t executable and configuration file | Start TRACE32 |
| C/C++ Postmortem | Parameters to launch TRACE3 | 2 | |
| Launch Group Launch Group (Deprecated) | TRACE32 executable: | C:\T32\bin\windows64\t32marm.exe | Browse |
| Target Communication Fra TRACE32 TCF | Configuration file: | C:\T32\config.t32 | Browse Edit |
| A New_configuration | Configuration file Parameters | : | |
| | Startup script: | C:\T32\demo\arm\compiler\gnu\demo_sram.cmm | Browse Edit |
| | Script Parameters: | | |
| | Working path: | C:\T32\demo\arm\compiler\gnu | Browse |
| Вн | Automatically start TRACE32 | with the debug configuration (TCF port number must be specified) | |
| _ | Target ID: | | |
| | Delay starting TCF (s): | | |
| | Command line: C:\T32\demo\ | arm\compiler\gnu: C:\T32\bin\windows64\t32marm.exe -c C:\T32\conf | ig.t32 -s C:\T32\demo\arm |
| C- | Start TRACE32 PowerView o | n remote host over Telnet or SSH. (The config file should contain SCREE | N=OFF) |
| | | | |
| Filter matched 9 of 9 items | | Re | vert Apply |
| ? | | | Close |

3. Click **Apply** when you are done.

| TRACE32 executable | Used TRACE32 executable e.g. C:\T32\bin\windows64\t32marm.exe |
|-------------------------------|--|
| Configuration file | Used TRACE32 configuration file e.g. C:\T32\config.t32. The configuration file must contain the TCF block as described in "Configuration File", page 24. |
| Configuration file parameters | Parameters for the configuration file |
| Start-up script | When a TRACE32 instance starts, the PRACTICE script autostart.cmm is executed, which then calls the following scripts: system-settings.cmm (from the TRACE32 system directory, usually C:\t32) user-settings.cmm (from the user settings directory: on Windows %APPDATA%\TRACE32 or ~/.trace32 otherwise) work-settings.cmm (from the current working directory) Here you can specify an additional PRACTICE script which is automatically started afterwards. |
| Script parameters | Parameters for the start-up script file |
| Working path | Active directory after starting the TRACE32 instance. |

[B] Start TRACE32 with the debug configuration

| Target ID | This option can be used only if TCF discovery in disabled. A new target with a specific port must be declared first (see Manual Debug Target Setup), then its "Target ID" should be written into this box. |
|-----------|--|
| Delay | An optional delay time until TRACE32 complete its startup. |

[C] Start TRACE32 on remote host

When checked, TRACE32 PowerView will be open in a remote machine using Telnet or SSH protocol.

The parameters specified in [A] should exists in the remote machine.

To configure TRACE32 as a TCF agent, you need to add the following lines to your TRACE32 configuration file. The default configuration file is config.t32 and is located in the TRACE32 system directory.

;T32 TCF Access
<- mandatory empty line
<- optional comment line
<- mandatory empty line
</pre>

An optional TCF port number can be added to the configuration file. If a port number is specified, then the TCF discovery mechanism is disabled in TRACE32. The TCF front-end (Eclipse) needs then to connect to TRACE32 using the specified port number. This will be explained in details later in this document.

| ;T32 TCF Access TCF= | <- mandatory empty line <- optional comment line |
|-------------------------|--|
| PORT=1534 | <- TCF discovery disabled <- mandatory empty line |



An empty line is mandatory before and after the TCF block in the TRACE32 configuration file, otherwise, a "syntax error" will be reported when starting TRACE32.

For more information about the TRACE32 configuration, please refer to "**Training Basic Debugging**" (training_debugger.pdf).

T32Start

In case you are using t32start.exe utility to start TRACE32, you can enable TCF under **Advanced Settings** > **Interfaces** > **TCF Port.** Please note that at least t32start.exe version 2.4.7 is required.



Please refer to the "T32Start" (app_t32start.pdf) manual for more information about the T32Start utility.

Start TRACE32

To start TRACE32 from within Eclipse:

- 1. Choose **Run** menu > **Debug Configurations**.
- 2. In the left window pane of the **Debug Configurations** window, click a configuration under the entry **TRACE32 TCF**.

| Debug Configurations | | | × |
|---|--|--|---|
| Create, manage, and run configu Connect to Lauterbach TRACE32 ove | urations er the Target Communication Fram | ework | Ť. |
| Image: Second system Image: Second system <td>Name: New_configuration Image: Target Download Image: Select Parameters to launch TRACE32 TRACE32 executable: Configuration file: Configuration file: Configuration file: Configuration file Parameters: Startup script: Script Parameters: Working path: Image: Automatically start TRACE32 Target ID: Image: Delay starting TCF (s): Image: Command line: C:\T32\demo\u00ex Comerview or the start TRACE32</td> <td>ymbol Files Beth Map Source Comm TRACE32 t executable and configuration file C C:\T32\bin\windows64\t32marm.exe C C:\T32\config.t32 C C:\T32\demo\arm\compiler\gnu\demo_sram.cmm C:\T32\demo\arm\compiler\gnu with the debug configuration (TCF port number must be specified) arm\compiler\gnu: C:\T32\bin\windows64\t32marm.exe - c remote host over Telnet or SSH. (The config file should contain SCREEN:</td> <td>Start TRACE32 Browse Browse Edit Browse Edit Browse Edit Comparison</td> | Name: New_configuration Image: Target Download Image: Select Parameters to launch TRACE32 TRACE32 executable: Configuration file: Configuration file: Configuration file: Configuration file Parameters: Startup script: Script Parameters: Working path: Image: Automatically start TRACE32 Target ID: Image: Delay starting TCF (s): Image: Command line: C:\T32\demo\u00ex Comerview or the start TRACE32 | ymbol Files Beth Map Source Comm TRACE32 t executable and configuration file C C:\T32\bin\windows64\t32marm.exe C C:\T32\config.t32 C C:\T32\demo\arm\compiler\gnu\demo_sram.cmm C:\T32\demo\arm\compiler\gnu with the debug configuration (TCF port number must be specified) arm\compiler\gnu: C:\T32\bin\windows64\t32marm.exe - c remote host over Telnet or SSH. (The config file should contain SCREEN: | Start TRACE32 Browse Browse Edit Browse Edit Browse Edit Comparison |
| ← Ⅲ ► Filter matched 9 of 9 items | | Reve | ert Apply |
| ? | | Del | bug Close |

- 3. Click the **TRACE32** tab.
- 4. Click the **Start TRACE32** button.

After starting, TRACE32 executes the PRACTICE start-up script (*.cmm) you have specified.

- 5. Have you specified a fixed TCF port number in the TRACE32 configuration file?
 - No: Please continue with section TCF Discovery.

- Yes: Please continue with section Manual Debug Target Setup.

If this button is not displayed in the tool bar, take these steps:

- Choose Windows menu > Perspective > Customize Perspective.
- On the **Tool Bar Visibility** tab, select the check box Lauterbach TRACE32.

To establish a debug connection using the TCF discovery:

1. In the **Debug Configurations** dialog, click **Target Communication Framework**.

| Debug Configurations | | | | |
|--|--|--|--|--------------------------|
| reate, manage, and run configurations Run or debug a program using Target Comm | unication Framework | | | Ś |
| * 🗈 🗙 🗖 🔅 ▾ | Name: New configuration | | | |
| | Target Sh Download | Application 🖓 Arguments 🖉 Environme | unt Ex Symphol Files & Dath Man & Saurea C | |
| C (/C++ Application C (/C++ Attach to Application C (/C++ Postmortem Debugger C (/C++ Remote Application C Target Communication Framework New_configuration | Run TCF symbols server Run instance of TCF age Use local host as the targ Target Target ID: TCP:12 Available targets: B Name B TACE32 Power | 7.0.0.1:54056 OS User /iew for ARM64 Windows 7 kjmal | Transp Host Port TCP 127.0.0.1 55529 | New |
| ter matched 7 of 7 items Peer ID Peer na | eer Properties : [TCP:127.0.0.1:55529 me: TRACE32 PowerViet | v for ARM64 0 [Power Debug II @ 192.10 | Revert 68.187.43] | Remove Apply Close |
| Peer pro | perties: | Value | | |
| ■ Ac | - ientID | 56ead1c4-bfb6-4a99-870e-2d762 | Add | |
| e Ho | ost | 127.0.0.1 | Remove | |
| ● 09 | Name | Windows 7 | | |
| e Po | rt | 55529 | | |
| 🔍 🔍 Se | rviceManagerID | 56ead1c4-bfb6-4a99-870e-2d762 | 2 | |
| 🗢 TR | ACE32 Hardware | PowerDebug-II | | |
| I TR | ACE32 Version | S.2016.03.000071216MXD | | |
| 🔍 🔍 Ta | rget Architecture | ARM64 | | |
| 🔍 Tr | ansportName | ТСР | | |
| 🔍 Us | erName | kjmal | | |
| Us | envame | kituai | | |

- A See Step 2. below.
- **B** If the TCF discovery has been enabled in TRACE32, then you should see the TRACE32 PowerView instance on the **Target** Tab under **Available Targets**.
- **C** A double-click on the **Name** will show the properties of the TRACE32 PowerView instance including the target architecture, the TRACE32 software version and the used TRACE32 hardware.
- 2. In case you are using the standard Target Communication Framework configuration, clear all three check boxes on the **Target** tab:
 - Run TCF symbols server on the local host
 - Run instance of TCF agent on the local host
 - Use local host as the target

3. To establish a debug connection, select the TRACE32 instance under **Available Targets**, and then click the **Debug** button.



If the TCF discovery is enabled in TRACE32, the TCF port number will be automatically selected. This means that a new port number could be used each time a new TRACE32 PowerView instance is used.

4. Choose **Windows** menu > **Perspective** > **Other** > **Debug**.

| Debug2 - c:\t32\demo\arm\compiler\gnu-pic\sieve.c | - Eclipse | | | | | • × |
|---|----------------------------------|---------|--------------------|------------------|--------------------|----------|
| File Edit Source Refactor Navigate Search Pro | ject Run TRACE32 Window | v Help | | | | |
| 🔍 🕪 II 🔳 🕅 3. 🖘 .R i> 🗮 📧 🗛 🗸 | ☆ - ○ - 🌯 - 🖻 🛱 | C/C++ | 🎄 Debug2 💠 | Debug | Quick Access | : 📝 💸 |
| 🎋 Debug 🖾 | 🍇 🕅 🖬 🔻 🗖 | | o Breakpoints 🔀 | Modules | | |
| New configuration (TRACE32 PowerView For AR) | M) | | | ¥ ‰ # | | - < |
| OMAP4430APP1 (Suspended: Signal: stopped | at breakpoint) | | 💷 💂 sieve s [: | | al Itune: Software | |
| 0x403017e8 [sieve_arm] main():emo\arn | n\compiler\gnu-pic\sieve.c, line | 650 | V Sieve c [| iunction: sievel | [type: Software] | 1 |
| 0x403000e0 [sieve_arm] gomain() | | | N B sievele [i | unction, sievej | (type: soltware) | |
| | | N | lo scope specified | | | |
| 🔝 crt0-pic.s 🔄 sieve.c 🔀 🚟 Disassembly | | | 해 Registers 🛛 | 🏝 📲 🖻 | 8 🖬 🖻 | ~ - 8 |
| 643 do { | | | Name | Hex | Decimal | Descr * |
| 644 if (monHook) | | | 1919 RO | fffffb96 | 4294966166 | |
| 645 monHook(); 646 if (watchdogTrigger) | | | 1910 R1 | 40302128 | 1076896040 | E |
| 647 watchdogTrigger(); | | | 1919 R2 | 000000c | 12 | |
| 648 | | | 1919 R3 | 40302118 | 1076896024 | |
| 649 mstatic1 = 12; | | | 1919 R4 | 40302068 | 1076895848 | |
| •3650 mstatic2 = 34; | | | 1919 R5 | 000020d0 | 8400 | |
| 652 | | | 1919 R6 | 2351db2e | 592567086 | |
| 653 inc = (4 * 15000) / period; | | | 1010 R7 | 40300000 | 1076887552 | |
| 654 sign = ((mcount % period) > | period/2) ? -1 : +1; | | 1010 R8 | 00000001 | 1 | - |
| 655 plot1 = plot1 + sign * inc; | | | • | | | Þ |
| 657 proc2 = 25000 · sign; | | _ | Hex: fffffb96 | Dec: 4294966 | 166 Oct: 037777 | 775626 |
| 658 vtriplearray[0][0][0] = 1; | | | Bin: 1111, 1111 | ,1111,1111, | 1111,1011,1001 | ,0110 |
| 659 vtriplearray[1][0][0] = 2; | | | Size: 4 bytes, re | adable, writable | | |
| 660 vtriplearray[0][1][0] = 3; | | | | | | |
| 661 vtripiearray[0][0][1] = 4; | | - | 1 | | | T |
| Concela Contesta Decklareza O Succetableza | (v)= Veriables [22] | P | the sets of | പറിക്ക | × | |
| Name | Type | | | : ≪ | 36 🖸 🖸 | |
| AA : | int int | | | - F | | |
| (X)=) | Int | | - | | | = |
| (x)= inc | short | | 1 | | | |
| (∧)- sign | char* | | | V/030210c | | |
| | | | | | | • |
| 60 | | | | | | * |
| Hex: 003c, Dec: 60, Oct: 074 | | | | | | T |
| | | | | | | • |
| W | ritable Smart Insert | 650 : 2 | 3 | | | |

If the TCF discovery has been disabled in TRACE32 by specifying a fixed TCF port number in the configuration file, then you need to create a new target setup.

To create a new target setup:

- 1. Choose **Run** menu > **Debug Configurations** dialog > **Target Communication Framework** to open the **Debug Configuration** window.
- 2. Click the New button on the Target tab.

The TCF Debug Target Setup dialog opens.



- 3. Select Manual setup of TCF connection properties, and then click Next.
- 4. In the **Peer name** field, enter any name e.g. "TRACE32"
- 5. For the **Port** property, set the port number used by TRACE32, e.g. we use here the default port number 1534, then press **Finish**.

| CF Debug | Target Setup | | |
|---------------|-------------------|---------------|--------|
| Manual clie | nt configuration | | Ť. |
| Peer ID: | USR:1458234684800 | | |
| Peer name: | TRACE32 | | |
| Peer properti | es: | | |
| Name | | Value | Add |
| Host | _ | 127.0.0.1 | |
| Port — | | 1534 | Remove |
| Transpo | ortName | TCP | |
| | | | |
| | | | |
| ? | < Back | Next > Finish | Cancel |

A new entry with the selected name will then appear on the **Target** tab under **Available Targets**, see screenshot below.

| Debug Configurations | | Barris and Street | | | | | | — × | 3 |
|--|------------------------|--|-------------------------|---------------|-------------------|--------------------------|----------|------------|---|
| Create, manage, and run configurations Run or debug a program using Target Commu | nication Framework | | | | | | | Ť | |
| Image: Second Secon | Name: New_configuratio | on ad Application Mar Argu er on the local host arget :1458235323136 OS | ments 🖾 Enviror User | Transp TCP | Host 127.0.0.1 | Path Map Port 1534 | Source C | ommon | |
| Filter matched 7 of 7 items | | | | | | | Revert | Apply | |
| ? | | | | | | | Debug | Close | |

- 6. To establish a debug connection, select the TRACE32 instance under **Available Targets**, and then click the **Debug** button.
- 7. Choose **Windows** menu > **Perspective** > **Other** > **Debug**.

| Debug2 - c:\t32\demo\arm\compiler\gnu-pic\sieve | .c - Eclipse | | | | | |
|---|---------------------------------|----------|--------------------|---------------------------------|--------------------------|----------|
| File Edit Source Refactor Navigate Search P | roject Run TRACE32 Wind | low Help | | | | |
| 🔍 🕩 II 🔳 🕅 3. 19. le ið 🗟 🗷 🗛 🗸 | 🎋 🕶 💽 🕶 🦉 😁 | C/C++ | 🎄 Debug2 🐐 | Debug | Quick Access | i 🗾 💱 |
| 🎄 Debug 😫 | 🍇 🕷 🚺 🖬 💎 🖞 | | 💁 Breakpoints 🖂 | Modules | | |
| New_configuration (TRACE32 PowerView For A | RM) | | | X 🗞 🎜 | ; 🔊 🗙 🖛 🖻 | 5 - |
| 🔺 🧬 OMAP4430APP1 (Suspended; Signal: stoppe | ed at breakpoint) | | 🔽 🎖 sieve.c [a | address: 403017 | e8] [type: Software] | |
| 0x403017e8 [sieve_arm] main():emo\a | rm\compiler\gnu-pic\sieve.c, li | ne 650 | 🔽 🧋 sieve.c [f | function: sieve] | [type: Software] | |
| 0x403000e0 [sieve_arm] gomain() | | - | | | | |
| | | | No scope specified | | | |
| 🔝 crt0-pic.s 📄 sieve.c 🔀 🎫 Disassembly | | | 1919 Registers 🔅 | s 🏝 📲 🖻 | 8 🖬 🖻 🕈 | ~ |
| 643 do { | | * | Name | Hex | Decimal | Descr 🔺 |
| 645 monHook(); | | | 1919 RO | fffffb96 | 4294966166 | |
| 646 if (watchdogTrigger) | | | 1919 R1 | 40302128 | 1076896040 | = |
| 647 watchdogTrigger(); | | | 1919 R2 | 000000c | 12 | |
| 648 649 mstatic1 = 12; | | | 1111 R3 | 40302118 | 1076896024 | |
| \$650 mstatic2 = 34: | | | 10101 R4 | 40302068 | 1076895848 | |
| 651 mcount++; | | | 1111 R5 | 000020d0 | 8400 | |
| 652 | | | 1919 R6 | 2351db2e | 592567086 | |
| 653 inc = (4 * 15000) / period | ; | | 1110 R7 | 40300000 | 1076887552 | |
| 655 nlot1 = nlot1 + sign * inc: | > period/2) ? -1 : +1; | | 1919 R8 | 00000001 | 1 | Ψ. |
| 656 plot2 = 25000 * sign; | | | • | | | • |
| 657 | | | Hex: fffffb96 | , Dec: 4294966 | 5166, Oct: 037777 | 775626 🔺 |
| 658 vtriplearray[0][0][0] = 1; | | | Bin: 1111, 1111 | 1,1111,1111, adabla unitable | 1111,1011,1001 | ,0110 |
| vtriplearray[1][0][0] = 2; | | | Size: 4 bytes, re | adable, writable | 1 | |
| 661 vtriplearray[0][0][1] = 4; | | - | | | | - |
| • | | • | • | | | • |
| 📮 Console 🧔 Tasks 🖹 Problems 🜔 Executable | s (x)= Variables 🖂 | | <u></u> # [| a 🔊 🖉 🛪 | × 🔌 📑 🖻 📑 | ~ - 8 |
| Name | Туре | | • | Value | | * |
| (×)= j | int | | 5 | 55 | | = |
| (×)= inc | short | | (| 50 | | |
| (×)= sign | short | | 1 | 1 | | |
| ▶ ▶ p | char * | | | 0x4030210c | | * |
| | | | | | | , |
| 60 Hex: 003c, Dec: 60, Oct: 074 | | | | | | ÷ |
| • | | | | | | F |
| | Writable Smart Insert | 650 : | 23 | | | |

To activate the Debug perspective when a program is launched, do the following:

- 1. Choose **Windows** menu > **Preferences**.
- 2. In the left window pane, click **Run/Debug** > **Perspectives**.
- Under **Open the associated perspective when launching**, select the **Always** option. This will cause the perspective associated with a program to become active whenever it is launched.

| Preferences | | | |
|--|--|------------------------------------|--------------------------------------|
| type filter text | Perspectives | | < → < > < |
| ▷ Java EE ▷ Java Persistence | These settings control perspectives for running open when launching or when an application s | and debuggir uspends. | ng. A perspective can automatically |
| ⊳ JavaScript ⊳ Maven ⊳ Mylyn | Open the associated perspective when launch Always Never Prompt | iing | |
| Oomph Plug-in Development Remote Systems | Open the associated perspective when an app | lication suspe | nds |
| ▲ Run/Debug Breakpoints Console | These settings assign perspectives to each appl "None" to indicate that a perspective should no | ication type or t be opened. | launcher and launch mode set. Select |
| External Tools → Launching OpenOCD Peripherals views Perspectives QEMU SEGGER J-Link String Substitution Target Communicat TCP/IP Monitor View Management View Performance > Server > Team > Team > Terminal Validation > Web > Web > Web Services > Team > Web | Application Types/Launchers: Image: Second Secon | Modes/Pers Debug: De Run: No | spectives: bug |
| > XML | | | Restore Defaults Apply |
| ? . | | | OK Cancel |

In addition to the TCF communication, it is possible to control TRACE32 PowerView via a second UDP/IP channel using the Remote API. The TRACE32 View [A] can be used for this purpose. To open this view go to **Window** > **Show View** > **Others** > **Debug** > **TRACE32**. The command line [B] can be used to execute TRACE32 and PRACTICE commands similar to the TRACE32 command line. TRACE32 PowerView messages are then printed in the AREA field [C]. A **TRACE32** menu is additionally added to the Eclipse menu bar and contains shortcuts for special TRACE32 commands.

If TRACE32 is started from the TRACE32 Tab with an API port specified and the TRACE32 View is open, it will connect automatically. You can also connect to TRACE32 using the **Connect** button [**D**].



TRACE32

TCF=(illegal command)

If you get this error message when starting TRACE32, then your TRACE32 version is too old and does not support TCF. You should use a TRACE32 version from February 2016 or newer.

To check your TRACE32 version, choose **Help** menu > **About TRACE32**.

Eclipse

No TRACE32 PowerView instance under "Available Targets"

Please make sure that TRACE32 PowerView has been started as TCF agent with enabled TCF discovery.

1. To check this, select the TRACE32 Help menu > About TRACE32.

Under Environment, you can see the used configuration file.

2. Click **edit** and check if the configuration file contains the TCF block.

If the TCF discovery has been disabled by using a fixed port number in the configuration file, the target setup needs to be done manually. Moreover, you can use the PRACTICE function **TCF.PORT()** in TRACE32 to print the used port number (requires TRACE32 build 71550 or newer):

PRINT TCF.PORT()

Cannot locate peer TCP:<ip>:<port>

Please check that TRACE32 PowerView has been started as TCF agent and that you are using the correct port number in Eclipse.

Export the TRACE32 System Information

Be sure to include detailed system information about your TRACE32 configuration.

1. To generate a system information report, choose TRACE32 > Help > Support > Systeminfo.

| Lauterbach Homepage | | | |
|---------------------|------------------------------|---|--------------------------------|
| Support > | 🔑 System Information | | |
| About TRACE32 | 😌 Update TRACE32 | | |
| | 🔼 Technical Support Contacts | | |
| | 🖂 Contact Lauterbach 💋 | Generate TRACE32 S | upport Information |
| | Press th | he following button to get help on how to | generate Support Information: |
| | Compa | any: Lauterbach | Department: |
| | Prefix: | | |
| | Firstna | me: Andrea | |
| | Surnan | ne: Martin | |
| | Street: | Altlaufstr. 40 | P.O. Box: |
| | City: | Hoehenkirchen-Siegertsbr. | ZIP Code: 85635 |
| | Countr | ry: Germany | |
| | Teleph | one: (+49) 8102-9876-555 | |
| | eMail: | andrea.martin@lauterbach.com | |
| | | | |
| | Produc | t: PowerTrace PX | |
| | Target | CPU: ARM940T | |
| | Hostsy | stem: Windows 10 V | |
| | Compi | iler: Arm | |
| | Realtin | neOS: Nono | Safe Mode: |
| | | Generate Support Information: | Save to Clipboard Save to File |

NOTE: Please help to speed up processing of your support request. By filling out the system information form completely and with correct data, you minimize the number of additional questions and clarification request e-mails we need to resolve your problem.

- 2. Preferred: click Save to File, and send the system information as an attachment to your e-mail.
- 3. Click **Save to Clipboard**, and then paste the system information into your e-mail.

Export the Eclipse Error Log

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

- 1. Choose Window menu > Show View > Error Log to open the Error Log view in Eclipse.
- 2. On the Error Log view tool bar, click the Export icon.
- 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. Choose Help menu > About Eclipse.
- 2. Click the button **Installation Details**.
- 3. Click the **Configuration** tab.
- 4. Click **Copy to Clipboard**.
- 5. Paste the clipboard content into your support mail to Lauterbach.

SYStem.TCFconfig

The **SYStem.TCFconfig** command group is used to define TCF-specific setups for debugging.

See also

SYStem.state

SYStem.TCFconfig.TASKCONTEXT

Enable/disable task contexts

| Format: | SYStem.TCFconfig.TASKCONTEXT [ON OFF] |
|--------------|---|
| Default: ON. | |
| ON | Operating system tasks are displayed in the Eclipse debug view as child contexts. |
| OFF | Operating system tasks are not displayed in the Eclipse debug view as contexts. Only the name of the current task is displayed for information. |