LAUTERBACH
DEVELOPMENT TOOLS

Virtual Targets User's Guide

MANUAL

Virtual Targets User's Guide

Virtual Targets User's Guide

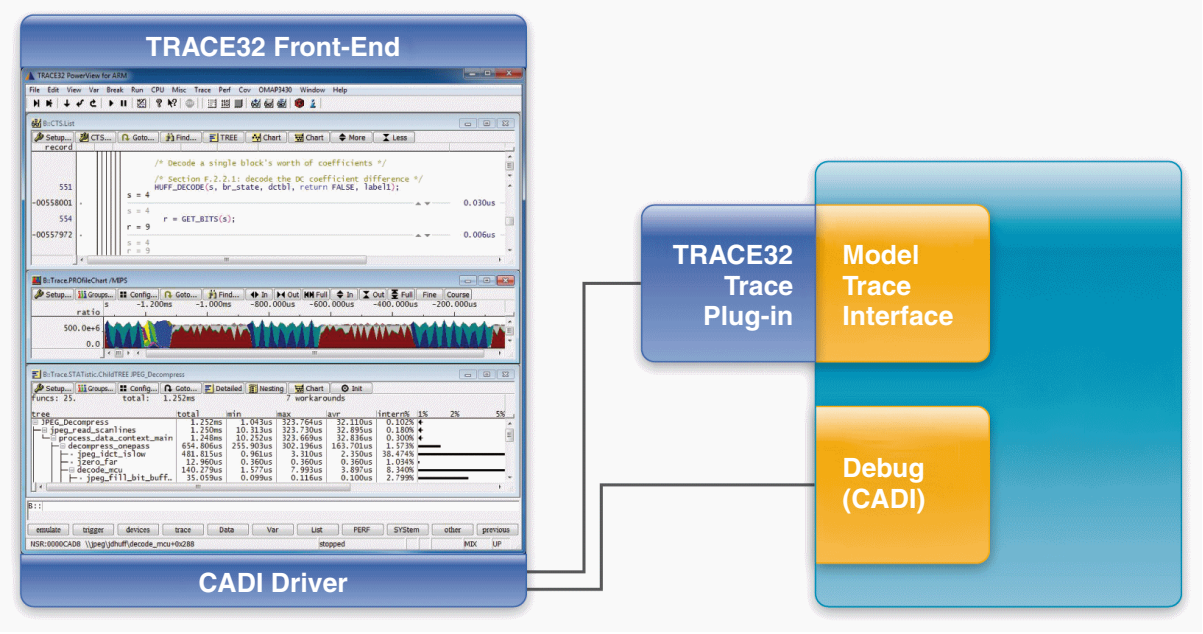
Version 06-Jun-2024

Introduction

Embedded system development is often time-critical. Waiting for a hardware prototype before software and test development can even be started may be unacceptable. Software models for the hardware can solve this problem. These software models of hardware prototypes are referred to as virtual prototypes or virtual targets.

The TRACE32 Debugging Front-End supports debugging and tracing of virtual targets. The short form *TRACE32 Front-End* will be used throughout this manual.

In this version, the manual describes how to install TRACE32 Front-End as a single-user installation, as well as how to configure and to start the application. It also describes how to use the TRACE32 PowerView GUI to connect to virtual targets via the CADI or MCD interface. For a list of all currently supported virtual targets, see **"Supported Virtual Targets"**, page 24.



Intended Audience

The audience of this manual are experienced Windows or Linux users who want to install and use the TRACE32 Front-End.

The degree of support that experienced Windows or Linux users may need from their administrators depends on factors such as:

- The license model selected for TRACE32 Front-End
- The installation and user rights granted to TRACE32 Front-End users by the operating system

How This Manual is Organized

- **Installing TRACE32 Front-End**: Describes how to install TRACE32 Front-End in Windows and Linux for the used license model.
- **Starting TRACE32 Front-End**: Describes how to start TRACE32 Front-End in Windows and Linux.
- **Connecting to Virtual Targets**: Describes how to connect TRACE32 Front-End to a virtual target.

Related Documents

This manual focusses on the single-user installation of TRACE32 Front-End. For information about multi-user installations, please refer to:

- **"TRACE32 Installation Guide"** (installation.pdf)

For information about license models, refer to:

- **"Floating Licenses"** (floatinglicenses.pdf).

For information about workarounds and solutions to common installation problems, please refer to:

- known_issues.txt - a file at the root level of the installation DVD.

For Windows only - The T32Start application assists you in setting up multicore/multiprocessor debug environments. For more information, please refer to:

- **"T32Start"** (app_t32start.pdf)

For information about TRACE32 Front-End on the Lauterbach website, please refer to:

- **https://www.lauterbach.com/frontend.html**
- **https://www.lauterbach.com/order/ord_frontend.html**
- **https://www.lauterbach.com/felist.html**

Contacting Support

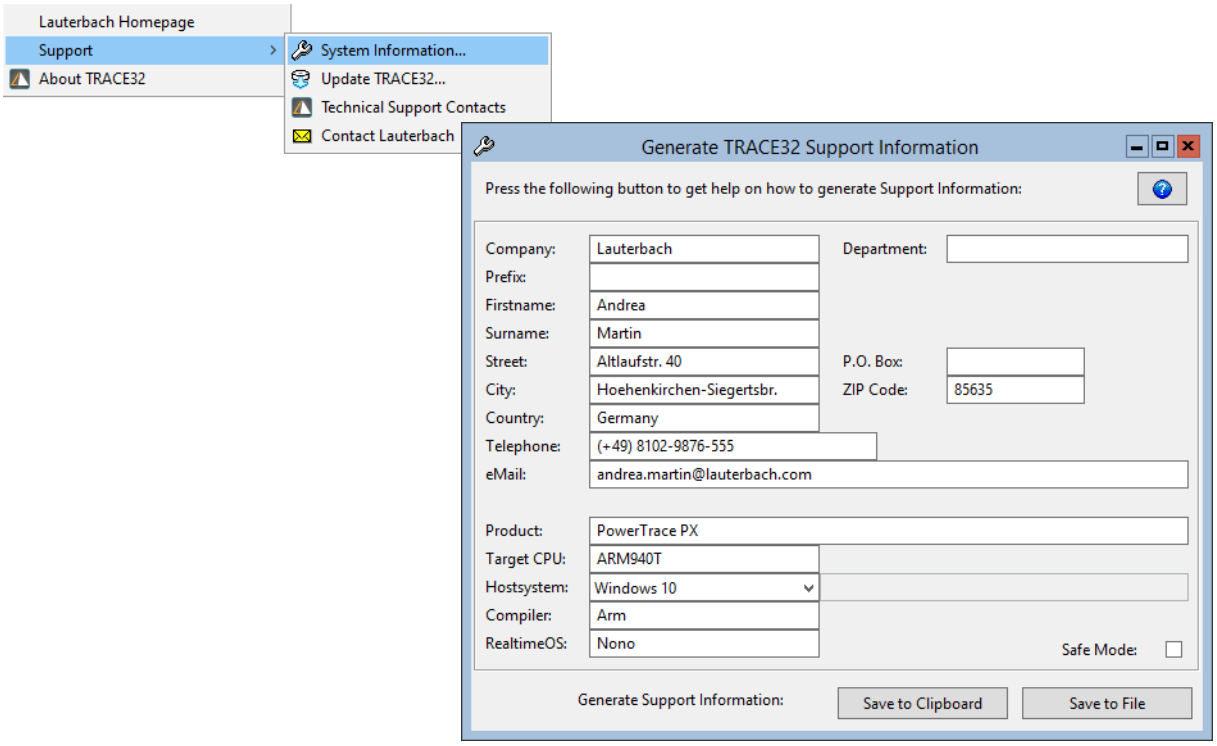
Use the Lauterbach Support Center: https://support.lauterbach.com

- To contact your local TRACE32 support team directly.
- To register and submit a support ticket to the TRACE32 global center.
- To log in and manage your support tickets.
- To benefit from the TRACE32 knowledgebase (FAQs, technical articles, tutorial videos) and our tips & tricks around debugging.

Or send an email in the traditional way to support@lauterbach.com.

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

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



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.

List of Abbreviations

CADI	Component Architecture Debug Interface (until Nov. 2011: Cycle Accurate Debug Interface)
GDB	GNU debugger
GDI	Generic debug instrument interface
GUI	Graphical user interface
MCD	Multicore debug
MDI	MIPS debug interface

Installing TRACE32 Front-End

This chapter describes how to install TRACE32 Front-End on the following operating systems:

1. **MS Windows**, see "**Installing TRACE32 Front-End in MS Windows**"
2. **Linux**, see "**Installing TRACE32 Front-End in Linux**"

Installing TRACE32 Front-End in MS Windows

An installer is available for the installation of TRACE32 Front-End in Windows.

System and other requirements:

- If one of the supported Windows operating system works fine on your computer, you can proceed to install TRACE32 Front-End.
- You need administrative privileges to install TRACE32 Front-End in MS Windows.
- Close all other programs before you start the installation.
- The following step-by-step installation procedure assumes that you already have a working TRACE32 installation on your computer and want to upgrade to TRACE32 Front-End.

Overview of the installation process:

The steps below are described in detail in the following sections.

- Install TRACE32 Front-End using the Windows installer.
- Modify the TRACE32 configuration file:
  - Check the **`PBI=`** settings.
  - Modify them if necessary.
- If floating licenses are used on Windows 32-bit clients, then install the license client DLL. Windows 64-bit clients are used, then skip this step.

NOTE:	We recommend that you make a backup copy of your TRACE32 installation before applying any TRACE32 software update.

To install TRACE32 Front-End in MS Windows:

1. Insert the Lauterbach TRACE32 installation DVD in your drive.
2. If the installation does not start automatically, navigate to the root level of the DVD.
3. Double-click setup.bat to start the installation process, and then follow the on-screen instructions.
4. If you have a previous installation of TRACE32, select the **Upgrade for new additional modules/OS** option.
5. On the **TRACE32 Product Type** screen, click **Software only (Simulator, Frontend, Native Debugger)**.
6. On the **Setup Type** screen, select the product type you want to install:
   - MCD Frontend
   - CADI Frontend
   - etc.
7. When you are prompted to insert path and file name of the interface library, do one of the following:
   - If you have your own interface *.dll, then click **Yes**.
   - If you do not have your own interface *.dll, then click **No**.
     If you click **No**, then the placeholder `example.dll` will be used in the config*.t32 file.
8. In the **CPU Selection** screen, select only the architectures you need. This speeds up the installation.
9. When you are prompted to choose between the 64-bit and 32-bit version, do one of the following:
   - If the used *.dll is a 64-bit version, click **Yes**.
   - Otherwise, click **No**.
10. In the screen **Prepare TRACE32 for Integration with other products** do one of the following:
    - If you want to use the GUI of TRACE32 Front-End for debugging virtual targets, then select the **No Integration** check box.
    - If you want to use the GUI of a tool other than TRACE32 Front-End for debugging virtual targets, then select one from the list.
11. Follow the on-screen instructions to complete the installation.

Modifying the Configuration Files in Windows

In a single-user installation, the location of the configuration files is: `%SYSTEMDRIVE%:\T32\`

Depending on the options selected during installation, one or several configuration files were installed.

Interface for virtual targets:	The installer generates these example configuration file:
CADI	configCADI.t32
GDI	configGDI.t32
MCD	configMCD.t32
MDI	configMDI.t32

To modify the configuration file for an interface:

1. Open the config*<interface>*.t32 file.
2. Adjust the `PBI=` setting for your single-user installation, see table and example below.
3. Save the file and close it.

PBI=	Specify the name of the virtual target by choosing one of the following interface types and library files:
	`CADI [<internal_library_file>]`
	`GDI <external_library_file>`
	`MCD <external_library_file>`
	`MDI`
	• The *<internal_library_file>* is provided by Lauterbach and is installed together with TRACE32. Specifying a library file (*.dll) is *optional* if it is enclosed in square brackets.
	• The *<external_library_file>* is not provided by Lauterbach. Please contact the manufacturer of the virtual target for the required *.dll file.

SCREEN=	Is the section heading; leave unchanged.
HEADER=	Title displayed in the main window of TRACE32 Front-End
PRINTER=	By default: `WINDOWS`

A line can be commented out in a configuration file using a semicolon.

The examples of the configuration files below reflect the single-user installation described in "**Installing TRACE32 Front-End in MS Windows Using the Installer**".

Example of the configMCD.t32 in Windows, i.e. for the MCD interface:	**Example of the configCADI.t32 in Windows, i.e. for the CADI interface:**

`PBI=MCD c:\vt-example\mcd.dll`
 	
`SCREEN=`
`HEADER=TRACE32`
 	
`; Printer settings`
`PRINTER=WINDOWS`
	`PBI=CADI`
	
	`SCREEN=`
	`; HEADER=TRACE32`
	
	`; Printer settings`
	`PRINTER=WINDOWS`

In the CADI configuration file, you can - *optionally* - specify the interface *.dll file:

- `PBI=CADI t32cadi.dll`

Installing the License Client DLL

If floating licenses are used on Windows 32-bit clients, install the license client DLL. Skip this step for Windows 64-bit clients.

To install the license client DLL:

1. Download the ZIP archive with the license client DLL (t32lm.dll) from **https://www.lauterbach.com/faq_license.html**
2. Unpack it and copy t32lm.dll into the TRACE32 system directory of your installation.

Installing TRACE32 Front-End in Linux

In Linux, the TRACE32 Front-End can be installed only manually. The steps below provide an overview of the entire installation process and are described in detail in the following sections.

- Install TRACE32 Front-End manually.
- Modify the configuration file.
- Create a desktop icon for TRACE32 Front-End.
- Install a PDF viewer in which you want to display the *.pdf files of the TRACE32 online help. For more information, see **SETUP.PDFViewer** in ide_ref.pdf.

Installing TRACE32 Front-End Manually in Linux

NOTE:	We recommend that you make a backup copy of your TRACE32 installation before applying any TRACE32 software update.

Two types of TRACE32 Front-End installations are possible in Linux:

- **Multi-user installation**: In this installation, TRACE32 Front-End is installed in /home/t32. For information about multi-user installations, refer to the **"TRACE32 Installation Guide"** (installation.pdf).
- **Single-user installation**: In this installation, TRACE32 Front-End is installed one level below the user's account in /home/*<account>*/t32.

```
/bin/
/boot/
/home/

# multi-user installation
/home/t32/…

# single-user installation
/home/<account>/t32/…
/media
…
```

By default, TRACE32 Front-End is installed in the /t32 directory, which is referred to as the *t32 system directory*. But installation locations other than /home/*<account>*/t32 are also possible. In either case, simply specify the installation location in the configuration file config.t32 after installing TRACE32 Front-End.

Installation prerequisites for single-user installations:

- Regular read and write permissions for your own *<account>* directory. The single-user installation of TRACE32 Front-End does not require any special user rights.
- Activating the bitmap fonts for Ubuntu or similar distributions and installation of Adobe Reader requires superuser rights.

TRACE32 Front-End can use Adobe Reader to provide a context-sensitive help system; i.e., type a command at the TRACE32 command line, press the spacebar and **F1**; the respective pdf file opens, displaying the description of the command.

The following step-by-step procedure explains a default single-user installation of TRACE32 Front-End in Linux (Ubuntu version 10.x). This installation is intended to serve only as an example.

Overview of the step-by-step procedure:

- Create the *t32 system directory* including subdirectories.
- Copy the files from the Lauterbach TRACE32 installation DVD to the newly created directories.

  The /home/*<account>*/plug_ins directory will then contain the trace32.api file. TRACE32 Front-End requires this file to communicate with Adobe Reader when you access the help system in a context-sensitive manner.
- Unzip and convert the files using the filecvt utility, which is included on the installation DVD.
- Download and install Adobe Reader.

The following assumptions are made:

- The name of the *<account>* is devuser
- The name of the installation DVD is TRACE32_20101110

Example - To install TRACE32 Front-End manually in Ubuntu Linux:

1. Insert the Lauterbach TRACE32 installation DVD in your drive.

   The installation DVD is automatically recognized by Linux, and the *<dvd_name>* is displayed in the file manager.

NOTE:	If the installation DVD is *not* automatically recognized, the following commands may be useful: • `ls /media` to determine the *<dvd_name>*. • `ls /dev` to list all the devices installed on your computer. • `df` to list all mounted media with free and allocated space. • `mount` to mount the installation DVD.

2. Open the terminal or shell in Linux.

3. Create the required directories:

```
mkdir /home/devuser/t32

mkdir /home/devuser/t32/bin/pc_linux

mkdir /home/devuser/t32/tmp

mkdir /home/devuser/plug_ins

cd /home/devuser/t32
```

4. Copy all files including the configuration file config.t32 to the newly created directories and assign write permissions:

```
cd /home/devuser/t32

cp -r /media/TRACE32_20101110/files/* .

cp -r /media/TRACE32_20101110/bin/pc_linux ./bin/pc_linux

cp /media/TRACE32_20101110/bin/pc_linux/config.t32 .

chmod -R u+w /home/devuser/t32/*
```

5. Unzip the *.gz files and move them to the /t32 directory:

```
/home/devuser/t32/bin/pc_linux/filecvt /home/devuser/t32
```

6. Copy the trace32.api plug-in to the /plug_ins directory:

```
cp /media/TRACE32_20101110/bin/pc_linux/trace32.api /home/devuser/
                                                              plug_ins
```

7. Enable the bitmap fonts for Ubuntu and similar distributions. Bitmap fonts are required for TRACE32.

```
cd /etc/fonts/conf.d
sudo ln -sf ../conf.avail/70-yes-bitmaps.conf /etc/fonts/conf.d
sudo rm -f 70-no-bitmaps.conf
```

Next: Modifying the Configuration File in Linux

Modifying the Configuration File in Linux

In a single-user installation, the name and location of the configuration file is:
/home/<*account*>/t32/config.t32

To modify the configuration file:

1. Open the configuration file.
2. Adjust the variables for your single-user installation, see table below.
3. Save the file and close it.

OS=	Is the section heading; leave unchanged.
SYS=	Installation location: Specify the path of the *t32 system directory*.
TMP=	Specify the path and directory for temporary files.
ID=	Enter the prefix `t32`

PBI=	Specify the name of the virtual target by choosing one of the following interface types and library files:
	`CADI [<internal_library_file>]`
	`GDI <external_library_file>`
	`MCD <external_library_file>`
	`MDI`
	• The *<internal_library_file>* is provided by Lauterbach and is installed together with TRACE32. Specifying a library file (*.so) is *optional* if it is enclosed in square brackets.
	• The *<external_library_file>* is not provided by Lauterbach. Please contact the manufacturer of the virtual target for the required *.so file.

The examples below reflects the single-user installation described in "**Installing TRACE32 Front-End in Linux**".

Example of the config.t32 in Linux - for the MCD interface:	**Example of the config.t32 in Linux - for the CADI interface:**

`OS=`
`SYS=/home/devuser/t32`
`TMP=/home/devuser/t32/tmp`
`ID=t32`
 	
`PBI=MCD` **`<mcd_interface>`**`.so`
	`OS=`
	`SYS=/home/devuser/t32`
	`TMP=/home/devuser/t32/tmp`
	`ID=t32`
	
	`PBI=CADI`

In the CADI configuration file, you can - *optionally* - specify the interface *.so file:

- `PBI=CADI t32cadi.so`

Desktop icons are referred to as launchers in Ubuntu Linux. The creation of a launcher for TRACE32 Front-End is an optional step, because you can also start the application via the terminal.

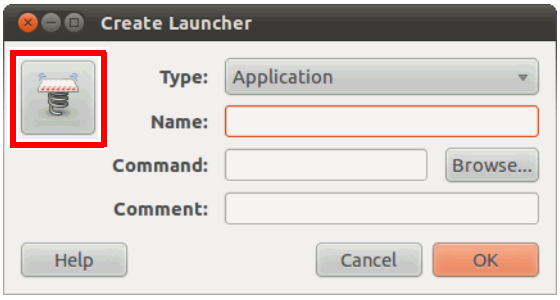
Parameters for the step-by-step procedure below:

Remember to replace *<account>* with your actual account name.

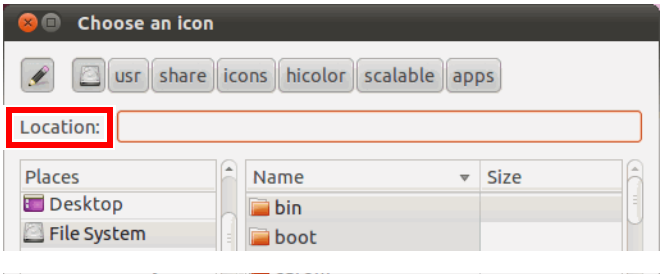
Icon location	`/home/<account>/t32/bin/pc_linux` For example, use this TRACE32 Front-End icon: trace32-48x48.png
Name box	TRACE32 Front-End
Command box	`/home/<account>/t32/bin/pc_linux/t32marm -c /home/<account>/t32/config.t32`

To create a launcher in Ubuntu Linux:

1. Right-click your desktop, and then select **Create Launcher**.



2. In the **Create Launcher** dialog, click the default icon.
3. In the **Choose an icon** dialog, browse for the icon location.



4. Select one of the icon files, and then click **Open** return to the **Create Launcher** dialog.
5. In the **Name** box, type the application name.
6. In the **Command** box, type the installation location.
7. Click **OK**.
   The new launcher appears on your desktop.

Starting TRACE32 Front-End

Many applications prompt you to enter the license key during the installation process. The behavior of TRACE32 Front-End is different because the behavior depends on the license model used.

Floating license	You are not prompted to enter a license (neither in Windows nor in Linux). TRACE32 Front-End checks the availability of floating licenses on the license management server: • If a license is available, it is automatically checked out from the license management server behind the scenes. TRACE32 Front-End is then ready for use. • If all licenses are in use, TRACE32 Front-End informs you that no floating license is available and remains in demo mode. • If the license management server for floating licenses is not configured correctly, TRACE32 Front-End remains in demo mode.

The following sections describe how to start (and license) TRACE32 Front-End.

Depending on the *operating system* used in your organization, choose the section relevant for you:

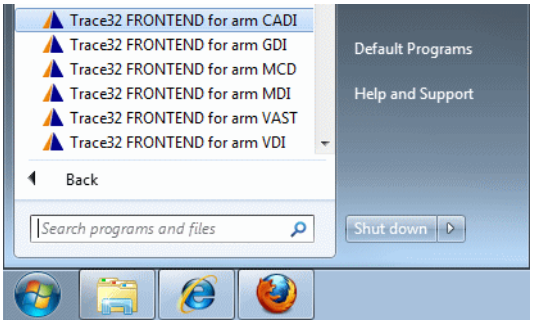
- "**Starting TRACE32 Front-End in Windows**"
- "**Starting TRACE32 Front-End in Linux**"

Starting TRACE32 Front-End in Windows

The following step-by-step procedure describes how to start TRACE32 Front-End with a floating license in Windows.

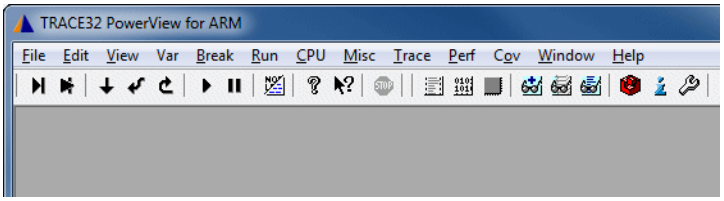
To start TRACE32 Front-End in Windows:

1. Click **Start** > **All Programs**.
2. Scroll through the list to locate **TRACE32**.
3. Click **TRACE32**, and then click the entry referring to the desired chip architecture; for example, **TRACE32 Front-End for arm CADI**.



Behind the scenes, TRACE32 Front-End checks out a license from the license management server if a license is available.

The main window of TRACE32 opens.



You are now ready to connect TRACE32 Front-End to a virtual target, see "**Connecting to Virtual Targets**".

Starting TRACE32 Front-End in Linux

The following step-by-step procedure describes how to start TRACE32 Front-End with a floating license in Linux.

To start TRACE32 Front-End in Linux - single-user installation:

1. Double-click the launcher if you have created one for TRACE32 Front-End.

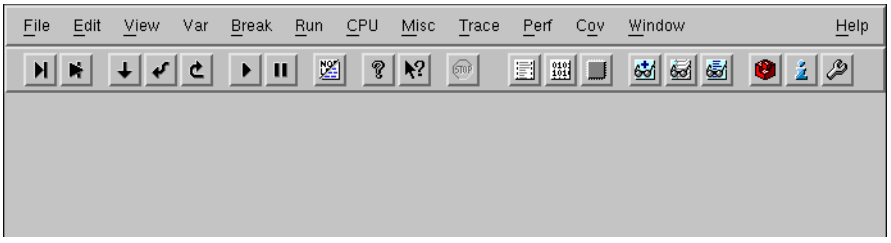
   Alternatively, open a terminal and enter these commands:

```
; Remember to replace <account> with your real account name.

cd /home/<account>/t32
/home/<account>/t32/bin/pc_linux/t32marm -c /home/<account>
                                                  /t32/config.t32
```

   Behind the scenes, TRACE32 Front-End checks out a license from the license management server if a license is available.

2. The main window of TRACE32 opens.



You are now ready to connect TRACE32 Front-End to a virtual target, see “**Connecting to Virtual Targets**”.

Connecting to Virtual Targets

This section applies to Windows and Linux. The steps described below are the same for all supported virtual targets.

NOTE:	Strictly observe the sequence of steps in the procedure below. Otherwise TRACE32 Front-End cannot connect to your virtual target.

To connect TRACE32 Front-End to a virtual target - Windows and Linux:

1. Start the virtual target.

   If TRACE32 Front-End was running before you started the virtual target, then close TRACE32 Front-End and start it again.

2. Start TRACE32 Front-End.

3. Specify the CPU.

SYStem.CPU *<cpu>*	Example of a *<cpu>*: CortexA5MPCore
SYStem.CONFIG.CoreNumber **CORE.NUMber**	Set up the number of cores you want TRACE32 to connect to (SMP system)
CORE.ASSIGN	Assign the cores

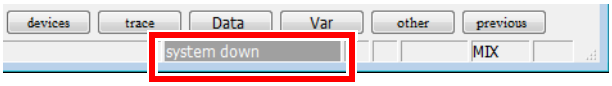
4. Specify the core you want to connect to.

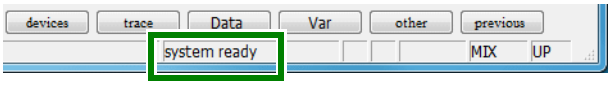
SYStem.CONFIG.ListCORE	Display the cores of a virtual target system
SYStem.CONFIG.CORE	Define the cores TRACE32 connects to in the virtual target

5. Enter debug mode:

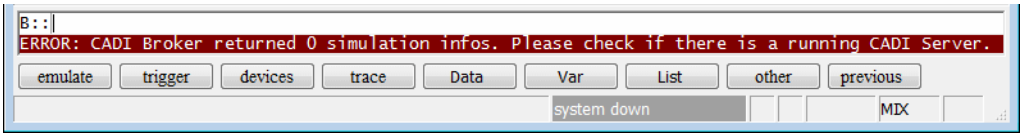
SYStem.Up

Result: The message in the status line changes from **system down** to **system ready**. TRACE32 Front-End is now connected to your virtual target.





If the connection cannot be established, then you receive an error message, which is shown below the command line. Example of such an error message:



Tracing Virtual Targets

This chapter describes how to configure trace support for TRACE32 Front-End and virtual targets.

GDI, MDI, and MCD

TRACE32 provides out-of-the-box trace support for the following virtual target interfaces: GDI, MDI, and MCD. Special settings to enable tracing are not required.

CADI

The trace data between the virtual target and TRACE32 is exchanged via a TCP/IP connection. Please ensure that the port to be used for tracing is not used by any other application. By default, the port for tracing is 21000.

Overview of the configuration process for the CADI interface

- Activate the TCP service.
- Modify the start of the virtual target.
- Inform TRACE32 about the port to be used for trace communication.

The following step-by-step procedures describe the configuration process in detail.

To activate the TCP service:

1. Open the TRACE32 config file. The default file name is configCADI.t32.
2. Add the following line to the config file:

```
TCPSERVER=      ;activates the TCP service
```

   This will activate the TCP service for TRACE32 and by default open the port 10000. This port is not important for tracing via the CADI interface. If required, this default port can be changed as follows:

```
TCPSERVER=
PORT=15000      ;changes the default port number from 10000 to 15000
```

3. Save the config file and close it.

To modify the start of the virtual target:

1. Pass the following command line argument to the command with which the virtual target is started:

   - **Case 1**: If the virtual target is a 32-bit version, then the 32-bit version of the CADI trace plug-in *must* be used.

     ```
     --plug-in "C:\T32\bin\windows\t32caditrace.dll"
     ```

   - **Case 2**: If the virtual target is a 64-bit version, then the 64-bit version of the CADI trace plug-in *must* be used.

     ```
     --plug-in "C:\T32\bin\windows64\t32caditrace.dll"
     ```

   The extended command will start the CADI trace plug-in together with the virtual target. CADI trace plug-in will open a TCP port; by default, port 21000.

2. To explicitly specify the port number, pass the following - optional - command line argument to the command with which the virtual target is started:

   ```
   --parameter TRACE.t32caditrace.PORT=22000
   ```

NOTE:	The TRACE.t32caditrace.PORT *must not* be the same as the `PORT=` setup in the TRACE32 config file.

3. Wait until the virtual target has fully started.

To inform TRACE32 about the port to be used for trace communication:

1. Start TRACE32 with the modified TRACE32 config file.
2. If you want to use a port other than the default port 21000, then inform TRACE32 about the TCP port to be used for trace communication:

   ```
   ;                                  <ip_address>    <port_number>
   SYStem.CADIconfig.Traceconfig       127.0.0.1         22000.
   ```

Result:

- The **Trace** commands such as **Trace.state** and **Trace.List** can now be used.

Supported Virtual Targets

For information about the supported virtual targets, visit:

https://www.lauterbach.com/felist.html