LAUTERBACH
DEVELOPMENT TOOLS

Sneakpeek Debug Back-End

MANUAL

Sneakpeek Debug Back-End

Sneakpeek Debug Back-End

Version 06-Jun-2024

History

08-Aug-2023 Initial version of this manual.

Introduction

The sneakpeek interface acts as a virtual debugport which faciliates the communication between a real target and the PowerView over TCP/IP. It is responsible for the initiation, translation and safe ending of the TCP/IP connection.

Related Documents

- **"T32Start"** (app_t32start.pdf): The T32Start application assists you in setting up multicore / multiprocessor debug environments, and software-only debug environments. T32Start is only available for Windows.

  For more information about software-only debug environments, please refer to: **"Software-only Debugging (Host MCI)"** (app_t32start.pdf).

Contacting Support

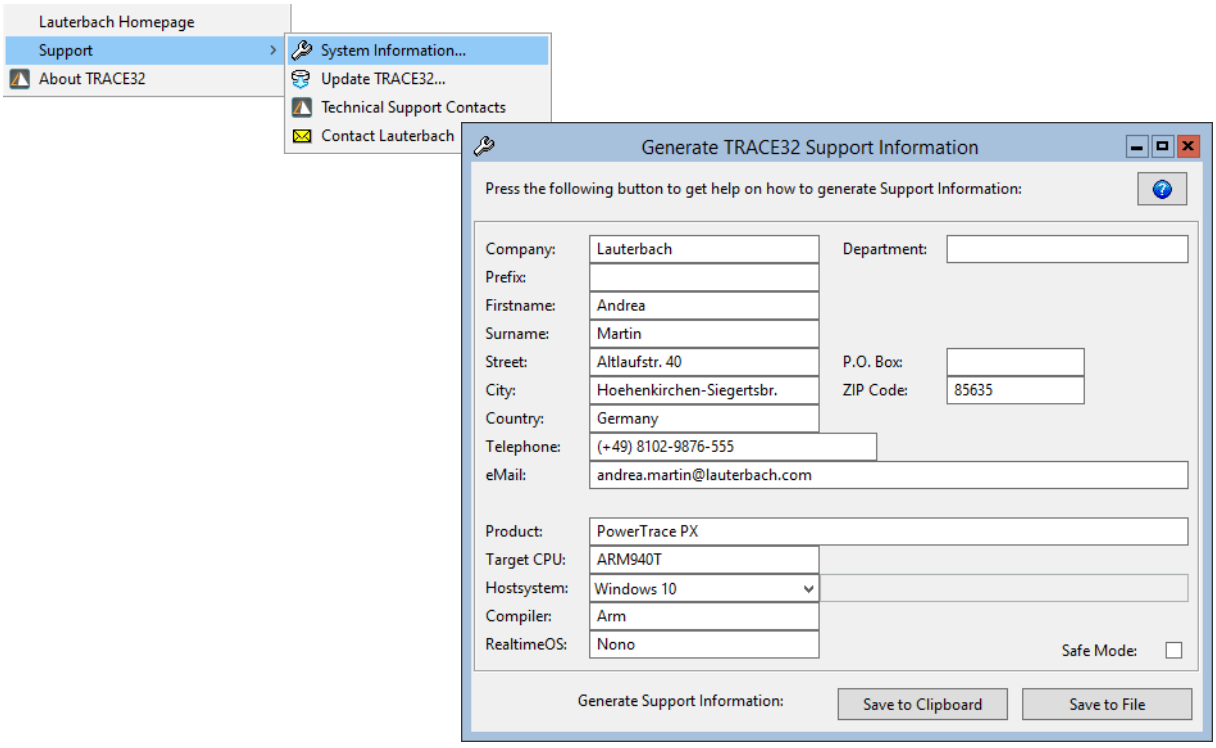
Use the Lauterbach Support Center: https://support.lauterbach.com

- 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**.

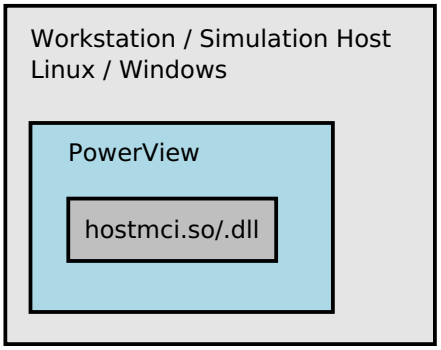


NOTE:	Please help to speed up processing of your support request. By filling out the system information form completely and with correct data, you minimize the number of additional questions and clarification request e-mails we need to resolve your problem.

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.

PowerView System Configurations

A single TRACE32 PowerView instance can be set up running on the same host as the back-end.



Modify the config.t32 file as follows:

```
PBI=MCILIB                          ; configure system to use hostmci.so
```

Linux example: To start TRACE32 PowerView with a specific config file, use e.g.:

```
bin/pc_linux/t32marm -c config.t32
```

For step-by-step instructions on how to configure the above setup in T32Start, see **"Debug Environment for Setup 1 (Single Instance)"** in T32Start, page 31 (app_t32start.pdf).

Configuring Sneakpeek

A typical startup sequence is shown below. It is a PRACTICE script (*.cmm, ASCII format) executed with the **DO** command.

```
; select SNEAKPEEK as back-end
SYStem.CONFIG.DEBUGPORT SNEAKPEEK0

SYStem.SNEAKPEEK.DISCONNECT

SYStem.SNEAKPEEK.MODELNAME   "MODEL_SNEAKPEEK"
SYStem.SNEAKPEEK.MODELCONFIG "NODE=192.168.0.3;PORT=7606;"

SYStem.SNEAKPEEK.LOG.AddOPTion SMARTCALLDATA
SYStem.SNEAKPEEK.LOG.OPEN "sneakpeek.log" /UNBUFFERED
SYStem.SNEAKPEEK.LOG.RemoveOPTion TOAREA

SYStem.CONFIG.AXINAME "AXI"
ON.ERROR CONTinue

; continue with CPU configuration
SYStem.CPU *
SYStem.CONFIG COREDEBUG BASE AXI:0xFF14010000
SYStem.CONFIG CTI       BASE AXI:0xFF14020000

SYStem.SNEAKPEEK.CONNECT
SYStem.PLOLLING DEFault OFF
MAP.UPDATEONCE 0x0--0xFFFFFFFFFFFFFFFF
SYStem.SNEAKPEEK.LOG.CLOSE

; connect to the emulation
SYStem.Up

ENDDO
```

SYStem.SNEAKPEEK — Configure SNEAKPEEK debug port

Using the **SYStem.SNEAKPEEK** command group, you can configure a SNEAKPEEK debug port. The command group is active after SNEAKPEEK has been selected as debug port. It allows to define and configure the used transactors and SNEAKPEEK 3rd-party library. The settings are shared among the TRACE32 instances connected to a certain MCI Server.

```
;optional step: open the SYStem.CONFIG dialog showing the DebugPort tab
SYStem.CONFIG.state /DebugPort

;selecting the SNEAKPEEK back-end activates the SYStem.SNEEKPEEK commands
SYStem.CONFIG.DEBUGPORT SNEAKPEEK0
```

See also

- SYStem.SNEAKPEEK.CONNECT
- SYStem.SNEAKPEEK.DISCONNECT
- SYStem.SNEAKPEEK.EXPLore
- SYStem.SNEAKPEEK.MODELCOMMAND
- SYStem.SNEAKPEEK.MODELCONFIG
- SYStem.SNEAKPEEK.MODELNAME
- SYStem.state

SYStem.SNEAKPEEK.CONNECT — Connect to emulation or simulation

Format: **SYStem.SNEAKPEEK.CONNECT** [**/TRY**]

Uses the settings previously configured with the **SYStem.SNEAKPEEK** commands to load the SNEAKPEEK library and connect to the emulation or simulation.

TRY — Forces the command to continue quietly when the connection could not be established.

See also

- SYStem.SNEAKPEEK

SYStem.SNEAKPEEK.DISCONNECT Disconnect from emulation/simulation

Format: **SYStem.SNEEKPEAK.DISCONNECT** ["*<transactor_name>*"] [**/UNUSED**]

Disconnects from existing connection to the emulation or simulation and disables the periodic re-connection tries.

<transactor_name> Disconnects a named transactor when it is not used anymore.

UNUSED Disconnects from all transactors that are not used anymore.

See also

- SYStem.SNEAKPEEK

SYStem.SNEAKPEEK.EXPLore Display plug-in capabilities

Format: **SYStem.SNEAKPEEK.EXPLore** [*<column>*]

<column>: **DEFault** | **Structure** | **Connected** | **tYpe** | **UsedByCommand** | **CoNFig**

The dialog can show the available transactor interface instances of the plug-in, provided the optional enumeration interface functions have been implemented by the plug-in.

DEFault Displays a pre-defined set of columns.

Structure Contains a tree with the abstractions layers of the SNEAKPEEK API. The top level enumerates all instances of the models or scenarios. The available transactor interface instances are displayed below the model.

Connected Displays whether TRACE32 has an active connection to a model or transactor instance. Mainly the commands **SYStem.SNEAKPEEK.CONNECT** and **SYStem.SNEAKPEEK.DISCONNECT** are used to change the connection state.

tYpe Type of the node, e.g. model or certain transactor type.

UsedByCommand Displays a list of configuration commands that are active and point to the transactor instance.

CoNFig Displays the configuration string.

Example:

```
SYStem.SNEAKPEEK.EXPLore DEFault
```

See also

- SYStem.SNEAKPEEK

SYStem.SNEAKPEEK.MODELCOMMAND Execute command in plug-in

Format: **SYStem.SNEAKPEEK.MODELCOMMAND** "*<command>*"

Executes a plug-in specific command.

Example:

```
SYStem.SNEAKPEEK.MODELCOMMAND "do something important"
LOCAL &result
&result=EVAL.STRing()
PRINT "Result was: &result"
```

See also

- SYStem.SNEAKPEEK

SYStem.SNEAKPEEK.MODELCONFIG Configure emulation options

Format: **SYStem.SNEAKPEEK.MODELCONFIG** "*<configuration>*"

Configures the options to connect to the emulation or simulator. The particular options are defined by the 3rd-party plug-in.

See also

- SYStem.SNEAKPEEK

Format: **SYStem.SNEAKPEEK.MODELNAME "***<model_name>***"**

Selects a certain emulation out of a set of emulations.

See also

- SYStem.SNEAKPEEK