



Tracing using DS-5 Debugger from the Command Line

Version 1.0

Non-Confidential

Copyright © 2018 Arm Limited (or its affiliates).
All rights reserved.

Issue 02

102650_0100_02_en



Tracing using DS-5 Debugger from the Command Line

Copyright © 2018 Arm Limited (or its affiliates). All rights reserved.

Release information

Document history

Issue	Date	Confidentiality	Change
0100-02	1 January 2018	Non-Confidential	First release

Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly

or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>.

Copyright © 2018 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Feedback

Arm® welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on <https://support.developer.arm.com>

To provide feedback on the document, fill the following survey: <https://developer.arm.com/documentation-feedback-survey>.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Contents

- 1. Overview..... 6
- 2. Trace Configuration in DS-5..... 7
- 3. DS-5 Command Prompt..... 8
- 4. Scripting the Debugger..... 10

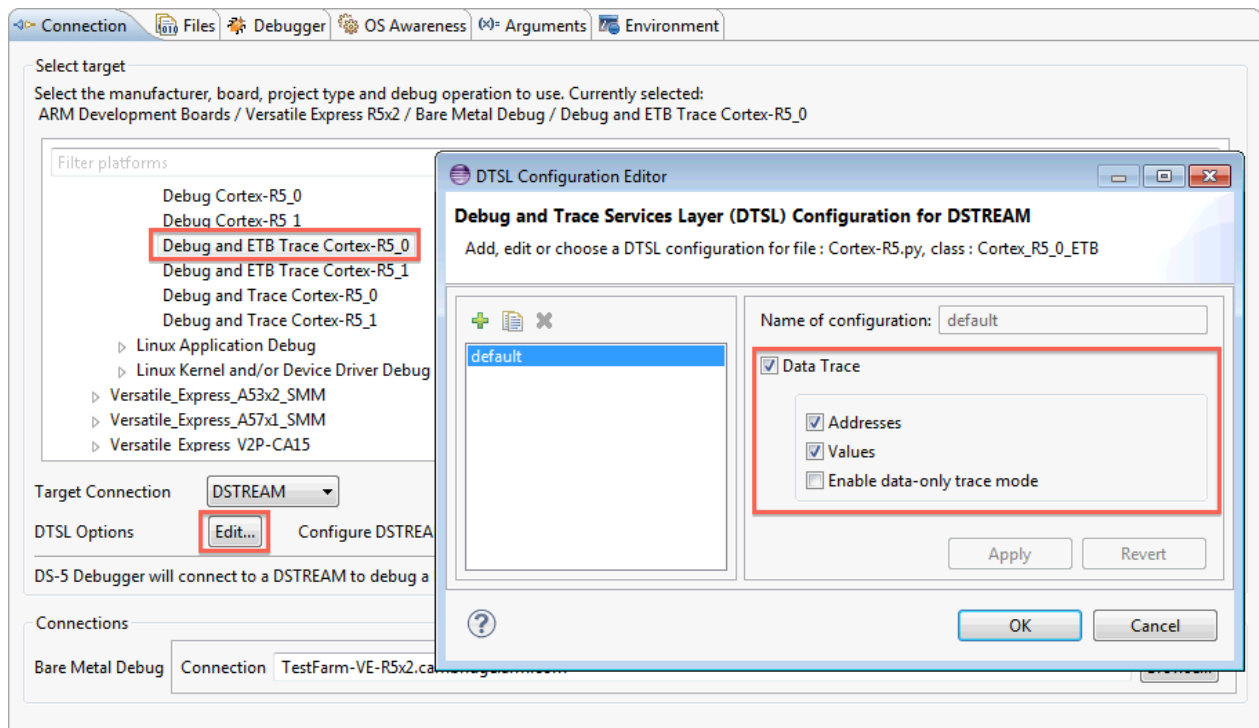
1. Overview

Sometimes it's useful to be able to drive DS-5 Debugger from the command line. If you are automating tests or carrying out repetitive tasks, this can help to save significant time. In this tutorial we're going to show you how to capture trace in this way.

2. Trace Configuration in DS-5

The first thing you need to do is make sure that you can capture trace using the graphical debugger. Open up Arm DS-5 Development Studio, switch to the DS-5 Debug perspective and open up the Debug Configuration for the device you are going to capture trace from.

Figure 2-1: Headless-trace-debug-configuration



Choose a connection type that enables trace. Some connections provide DTSL options to configure the trace capture. If this is the case with your connection, you should review them by clicking Edit...

Take a note of the specified class, in this case Cortex_R5_0_ETB. This is a clue to where the DTSL file is located, which you'll need later on.

What is DTSL?

DS-5 abstracts away a lot of the low-level complexity of connecting to a target by using Debug and Trace Services Layer (DTSL) scripts. For most off-the-shelf devices, you don't need to make any changes to these scripts. If you're developing a custom SoC, you can edit the DTSL script to match the configuration of the trace elements in your design.

3. DS-5 Command Prompt

This chapter shows the instructions which need to be done in Command Prompt.

Connecting to your Device

The component features of DS-5 can be run from the DS-5 Command Prompt. Open it up and then type `debugger` without any arguments to launch the help menu.

Specify a connection using `--cdb-entry`, followed by the details of the connection, which will take the format of “manufacturer::board::project type::debug operation::debug of specific core::debug hardware”.

If you're unsure about this, just type `--cdb-entry` with no options, which will display the list of devices in the database.

Once you have connected successfully, the debugger command prompt will appear. Type `quit` to exit.

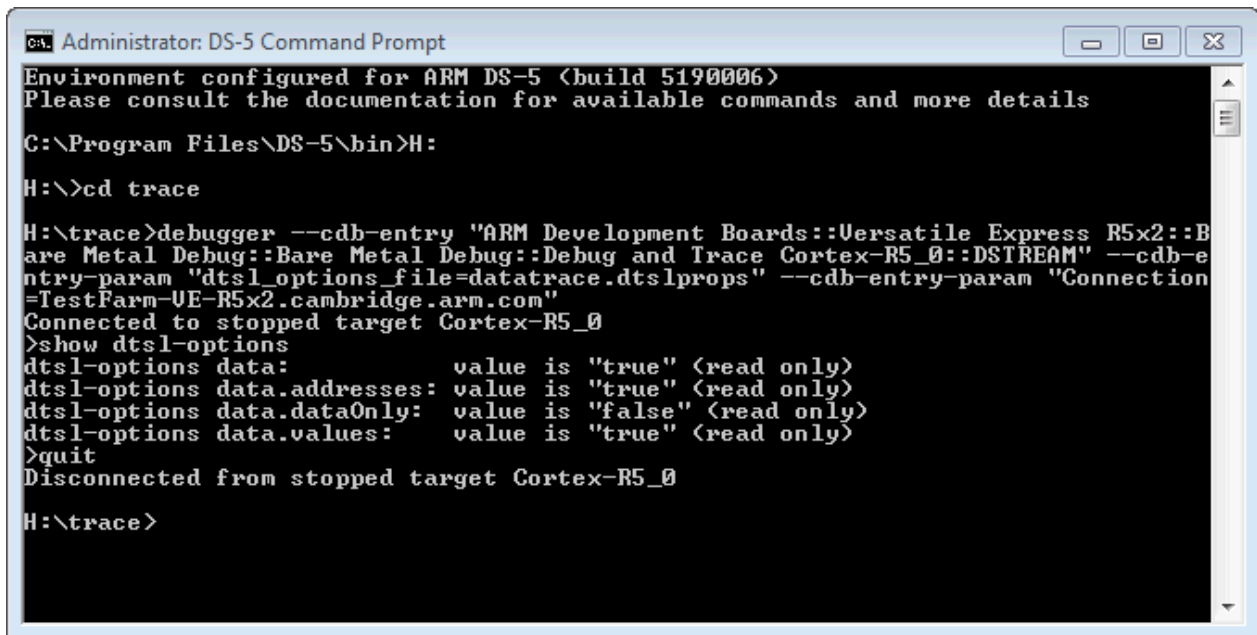
Adding DTSL Options

The DTSL options which are needed to configure trace are stored in a file with a `.dtslprops` extension. In our case, that file is located at: `C:\Documents\DS-5 Workspace\metadata\plugins\com.arm.ds\DTSL\ARM+Development+Boards+-+Versatile+Express+R5x2\Cortex-R5\Cortex_R5_0_ETB\default.dtslprops`.

Copy the path and then pass it to the debugger: `--cdb-entry-param="dtsl_options_file=C:\Documents\DS-5 Workspace\metadata\plugins\com.arm.ds\DTSL\ARM+Development+Boards+-+Versatile+Express+R5x2\Cortex-R5\Cortex_R5_0_ETB\default.dtslprops"`, replacing the file location with your own.

To specify which DSTREAM to connect to, enter: `--cdb-entry-param "Connection=x"` where `x` takes host name of the DSTREAM.

Here is a full command line showing connection to a target, confirming that the DTSL options are set:

Figure 3-1: Command prompt

```
C:\> Administrator: DS-5 Command Prompt

Environment configured for ARM DS-5 <build 51900006>
Please consult the documentation for available commands and more details

C:\Program Files\DS-5\bin>H:

H:\>cd trace

H:\trace>debugger --cdb-entry "ARM Development Boards::Versatile Express R5x2::Bare Metal Debug::Bare Metal Debug::Debug and Trace Cortex-R5_0::DSTREAM" --cdb-entry-param "dtssl_options_file=datatrace.dtsslprops" --cdb-entry-param "Connection=TestFarm-UE-R5x2.cambridge.arm.com"
Connected to stopped target Cortex-R5_0
>show dtssl-options
dtssl-options data:          value is "true" <read only>
dtssl-options data.addresses: value is "true" <read only>
dtssl-options data.dataOnly:  value is "false" <read only>
dtssl-options data.values:    value is "true" <read only>
>quit
Disconnected from stopped target Cortex-R5_0

H:\trace>
```

4. Scripting the Debugger

To automate the debugger in order to capture trace, you need to write a script for it to follow. The easiest way to do this is to start a debug session in the graphical debugger and carry out all the actions that you want to automate. These can then be exported directly from the Commands view.



You will need to manually add the `trace start` and `trace stop` commands as DS-5 Debugger doesn't automatically echo them to the Commands view.

Here's an example of a debugger script, which loads an image, enables trace, runs to a symbol called `end`, stops trace and then dumps the decoded trace to a text file:

```
show dtssl-options
loadfile image.axf
start
wait
trace start
advance end
wait
trace stop
trace report FILE=report.txt
quit
```

For more information on the `trace report` and `trace dump` commands, read the DS-5 documentation.

This should be everything you need to start capturing trace from the command line in DS-5. You can now automate your testing!