Using Trace Compass with CodeWarrior for ARMv8

1 Introduction

This document describes how to use the Trace Compass feature from CodeWarrior for ARMv8 release. **Trace Compass** is an open source toolkit that integrates open source trace frameworks/toolkits. It is based on Eclipse plugins (views) and binaries, and shared libraries. The trace compass tool allows you view and analyze trace data in various forms, like views, graphs, and metrics. The tool helps in extracting the useful information from traces in a much simpler and userfriendly manner.

NOTE

The Eclipse plugin is the architecture and Operating System agnostic based on a Java implementation. The C binaries are Operating System dependent. All of them can run only on a Linux-based system.

This application note includes the following sections:

- Setup requirements
- Collecting trace data using LTTng command-line tools
- · Importing trace data

2 Setup requirements

Contents

| 1 | Introduction | 1 |
|---|--|---|
| 2 | Setup requirements | 1 |
| 3 | Collecting trace data using LTTng command-line tools | 2 |
| 4 | Importing trace data | |



Collecting trace data using LTTng command-line tools

For the SDK based on DASH, before proceeding the trace collection, you have to enable LTTng within kernel and associated modules.

Ensure that LTTng layer is available and is compiled successfully, using the following commands:

1. Set CONFIG_BUILD_LTTNG to "y", replacing default "n" in

flexbuild/configs/build_lsdk.cfg

2. Build linux kernel

flex-builder -c linux
3. Build LTTng modules

- flex-builder -c lttng-modules
- 4. Prepare boot partition

flex-builder -i mkbootpartition

5. Prepare Ubuntu rootfs

flex-builder -i mkrfs -a arm64

6. Merge modules into rootfs and obtain its compressed .tgz version

flex-builder -i merge-component flex-builder -i compressrfs

7. Prepare board with the boot partition and rootfs images. For steps to obtain boot partition and rootfs images, refer LSDK documentation or use the flex-installer utility Help menu.

3 Collecting trace data using LTTng command-line tools

After the built image is boot up, you can start a trace session using LTTng toolkit. Perform the following steps to start the trace collection:

- 1. Create a trace session using the create command. By default, the traces are written in the root directory.
- 2. Filter the trace session by enabling only certain events/functions.
- 3. Start the trace session after all the settings are applied, using the start command. The stop command stops the tracing.
- 4. View the generated trace data using the view command, which is by default calling *babeltrace*.
- 5. Use Babeltrace for further processing of the trace log. You need to specify the trace path.
- 6. Destroy the trace session, using the destroy command followed by the name of the session.

The following list of commands shows an example of how to collect trace data using LTTng commands.

```
root@Ubuntu:~# lttng create mySession
Session mySession created.
Traces will be written in /root/lttng-traces/mySession-20171124-231444
root@Ubuntu:~# lttng enable-event sched switch -k
Kernel event sched switch created in channel channel0
root@Ubuntu:~# lttng start
Tracing started for session mySession
root@Ubuntu:~# lttng stop
Waiting for data availability....
Tracing stopped for session mySession
root@Ubuntu:~# lttng view
[23:15:20.516894700] (+?.???????) Ubuntu.ls1088ardb sched switch: { cpu id = 2 },
{ prev_comm = "swapper/2", prev_tid = 0, prev_prio = 20, prev_state = 0, next_comm = "lttng-
consumerd", next tid = 3812, next prio = 20 }
[23:15:20.518855700] (+0.001961000) Ubuntu.ls1088ardb sched switch: { cpu id = 5 },
{ prev_comm = "swapper/5", prev_tid = 0, prev_prio = 20, prev_state = 0, next_comm =
"rcu_preempt", next_tid = 8, next_prio = 20 } [23:15:20.518868540] (+0.000012840) Ubuntu.ls1088ardb sched_switch: { cpu_id = 0 },
{ prev_comm = "swapper/0", prev_tid = 0, prev_prio = 20, prev_state = 0, next_comm =
```

Importing trace data

```
"rcu sched", next tid = 9, next prio = 20 }
[23:15:20.518895420] (+0.000026880) Ubuntu.ls1088ardb sched_switch: { cpu_id = 5 },
{ prev_comm = "rcu_preempt", prev_tid = 8, prev_prio = 20, prev_state = 1026, next_comm =
"swapper/5", next_tid = 0, next_prio = 20 }
[23:15:20.518895980] (+0.000000560) Ubuntu.ls1088ardb sched switch: { cpu id = 0 },
{ prev comm = "rcu sched", prev tid = 9, prev prio = 20, prev state = 1026, next comm =
"kworker/0:1", next_tid = 111, next_prio = 20 }
[23:15:20.518923380] (+0.000027400) Ubuntu.ls1088ardb sched_switch: { cpu_id = 0 },
{ prev_comm = "kworker/0:1", prev_tid = 111, prev_prio = 20, prev_state = 1026, next_comm =
"swapper/0", next tid = 0, next prio = 20
[23:15:20.522714620] (+0.003791240) Ubuntu.ls1088ardb sched switch: { cpu id = 0 },
{ prev_comm = "swapper/0", prev_tid = 0, prev_prio = 20, prev_state = 0, next_comm = "mmcqd/
0", next tid = 1582, next prio = 20 }
[23:15:20.522731620] (+0.000017000) Ubuntu.ls1088ardb sched switch: { cpu id = 2 },
{ prev_comm = "lttng-consumerd", prev_tid = 3812, prev_prio = 20, prev_state = 2, next_comm
 "swapper/2", next_tid = 0, next_prio = 20 }
[23:15:20.522769420] (+0.000037800) Ubuntu.ls1088ardb sched_switch: { cpu_id = 0 },
{ prev_comm = "mmcqd/0", prev_tid = 1582, prev_prio = 20, prev_state = 1, next_comm =
"swapper/0", next tid = 0, next prio = 20 }
. . .
root@Ubuntu:~# lttng destroy mySession
Session mySession destroyed
root@Ubuntu:~#
```

4 Importing trace data

After collecting the trace data on the board using LTTng command-line utilities.

NOTE

Before importing and viewing the trace data, open the **Project Explorer** view and switch to **LTTng Kernel** perspective.

Perform the following steps to import a tracing project in the CodeWarrior:

1. Select **File > New > Project**.

The **New Project** wizard appears.

- 2. Expand the Tracing folder and select Tracing Project.
- 3. Click Next.

The Tracing Project page appears.

Importing trace data

| 🥦 Tracing Project | - • × | | | |
|---|-----------------|--|--|--|
| Tracing Project | | | | |
| | | | | |
| Project name: | | | | |
| ✓ Use <u>d</u> efault location | | | | |
| Location: C:\Users\b34930\workspace | B <u>r</u> owse | | | |
| Choose file system: default 💌 | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| (?) < <u>B</u> ack <u>N</u> ext > <u>Finish</u> | Cancel | | | |

Figure 1. Tracing Project wizard

4. Specify the **Project name** and click **Finish**.

The new project is added in the Projects view area.



Figure 2. Project View

5. Expand the node of the selected project. It contains the following folders:

- **Experiments** folder contains a set of traces that needs to be analyzed. These traces are selected from *Traces* folder. It displays several trace files by overlapping the data in the user interface.
- Traces folder contains all the imported trace files.
- 6. Right-click the **Traces** folder and select **Import** from the menu. The **Trace Import** wizard appears.

| 🥬 Trace Import | — | |
|---|---|-----------------|
| File system Source must not be empty | | Ľ |
| Select root directory: Select archive file: | | B <u>r</u> owse |
| Trace Type: <automatic detection=""> Import unrecognized traces Into folder: /Test_Import/Traces Contions</automatic> | | ~ |
| Overwrite existing trace without warning Create links in workspace Preserve folder structure Create experiment | | |
| ? <u>Einis</u> | h | Cancel |

Figure 3. Trace Import wizard

- 7. Browse the location of the **Source directory** that contains the tracing session folder.
- 8. Expand the node of the tracing session folder and select the trace folder.
- 9. Select the type of trace using **Trace type** field and click **Finish**.

The folder that contains the trace data gets added under the Traces folder of the Projects view.

Using Trace Compass with CodeWarrior for ARMv8, Rev. 11.3.2, 07/2018

Importing trace data

Double-click the trace file to access the trace data. You can now use all the advantages and features offered by Trace Compass toolkit.

| LTTng Kernel - Tracing/Traces/auto-20150518-143 | 959/kernel/kernel G | odeWarrior D | levelopment Studio fi | r QorIQ LS series - ARM V8 ISA | | | | | | | | - C -X |
|---|--|----------------|-----------------------|--------------------------------|--|------------------------------|--------------------------|----------------|--------------------|-----------------------------|------------------------|----------------------|
| Eile Edit Navigate Search Project Run Win | dow Help | | | | | | | | | | | |
| 📑 • 🗷 🗞 🍐 🗟 💁 • 😂 🛷 • 🛃 • | $\{1,\star\in\varphi,\star,\star\}$ | > - | | | | | | | | Quick Access | 🔡 🗟 C/C++ 🕇 | 🛼 LTTng Kernel |
| Project Explorer 😫 😐 🗖 | - Control Flow 23 | E Resource | es 🔲 Statistics | | | | | | | 🏇 💷 🖄 🖗 🖗 | ଜ 🐥 🔍 🔍 🗙 | , rþ 🔶 🗆 🗆 |
| 🖻 😫 😰 🍸 | Process | TID PTID | Birth time | Trace | | ▲ 17:40:40.470 1 | 17:40:40.480 17:40:40.49 | 0 17:40:40.500 | 17:40:40.510 17:40 | 40.520 17:40:40.530 17:40:4 | 0.540 17:40:40.550 | <u>^</u> |
| b District LinuxTraceDemo | init | 1 | 17:40:42.924309620 | auto-20150518-143959/kernel | | | | | | | | |
| a 🐸 Tracing | ksoftirad/0 | 3 | 17:40:40.509270460 | auto-20150518-143959/kernel | | | | | | | | |
| Experiments [0] | rcu preempt | 7 | 17:40:40.469295900 | auto-20150518-143959/kernel | | | | | | | | |
| a 🤔 Traces [2] | watchdog/0 | 11 | 17:40:44.319267260 | auto-20150518-143959/kernel | | | | | | | | |
| auto-20150518-143959 [1] | khugepaged | 340 | 17:40:41.169265260 | auto-20150518-143959/kernel | | | | | | | | |
| kernel | kworker/u16:2 | 746 | 17:40:44.599266740 | auto-20150518-143959/kernel | | | | | | | | |
| mySession-20150518-140104 [1] | kworker/0:3 | 844 | 17:40:40.469389140 | auto-20150518-143959/kernel | | | | | | | | |
| kernel | sh | 1314 | 17:40:40.486446180 | auto-20150518-143959/kernel | | | , i | | | | | |
| - | Ittng-sessiond | 1426 | 17:40:40.469445900 | auto-20150518-143959/kernel | | | | | | | | |
| | lttng-consumers | 1 1441 | 17:40:40.469305740 | auto-20150518-143959/kernel | | | | | | | | |
| | Ittng | 1464 | 17:40:40.485989700 | auto-20150518-143959/kernel | | | 1 | | | | | |
| | Ittng | 1471 | 17:40:52.510630620 | auto-20150518-143959/kernel | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | * |
| | | | | | | v 4 🔛 | | | | | | ÷. |
| | 💽 main.c 🛛 🤌 L | .inuxTraceDer | no.csv 🗄 auto- | 0150518-143959/kernel 😂 | | | | | | | | |
| | Timestamo | Char | nel Turne | Content | | | | | | | | |
| | A | Criar | ingi iyyye | content | | | | | | | | |
| - | A <srch></srch> | < srci | i> <srch></srch> | <srch></srch> | | | | | | | | |
| | 17:40:40.469 285 900 channel0_0 sched_switch prev_comm=lttng-sessiond, prev_tid=1426, prev_prio=20, prev_state=1024, next_comm=rcu_preempt, next_tid=7, next_prio=20 | | | | | | | | | | | |
| | 17:40:40.469 305 740 channel0_0 sched_switch prev_comm=rcu_preempt, prev_tida7, prev_prio=20, prev_tatae=1, next_comm=lttng-consumerd, next_tid=1441, next_prio=20 | | | | | | | | | | | |
| | 174040.469 389 140 channe0_0 sched_switch prev_comm=lttng-consumerd, prev_tid=1441, prev_prio=20, prev_state=0, next_comm=kworker/0.3, next_tid=844, next_prio | | | | | | | | | | | |
| | 17:40:40.469 440 980 channel0_0 sched_switch prev_comm=kworker/03, prev_tid=844, prev_prio=20, prev_state=1, next_comm=lttng-consumerd, next_tid=1441, next_prio | | | | | | | | | | | |
| | 17:40:40.469 445 900 channel0_0 sched_switch prev_comm=lttng-consumerd, prev_tid=1441, prev_prio=20, prev_state=2, next_comm=lttng-sessiond, next_tid=1426, next_p | | | | | | | | | | | |
| | 17:40:40.470 998 620 channe0_0 sched_switch prev_comm=lttng-sessiond, prev_tid=1426, prev_prio=20, prev_state=1024, next_comma.kworker/0:3, next_tid=844, next_prio | | | | | | | | | | | |
| | 17:40:40.471 019 700 channel0_0 sched_switch prev_comm=kworker/0-3, prev_tid=844, prev_prio=20, prev_state=1, next_comm=lttng-consumerd, next_tid=1441, next_prio | | | | | | | | | | | |
| | 1740/40.471 040 580 channel0_0 sched_switch prev_comm=lttng-consumerd, prev_tid=1441, prev_prio=20, prev_state=1, next_comm=lttng-sessiond, next_tid=1426, next_p | | | | | | | | | | | |
| | 17:40:40.479 28 | 6 060 chan | nel0_0_sched_switch | prev_comm=lttng-sessiond, | prev tid=1426, prev prio=20, prev state=1024, next | omm=rcu preempt, next tid= | 7, next_prio=20 | | | | | |
| | 17:40:40.479 25 | 2 340 chan | nel0 0 sched switch | prev commarcu preempt pr | rev tid=7, prev prio=20, prev state=1, next comm=1 | tno-sessiond next tid=1426 r | next prio=20 | | | | | |
| | 17:40:40.485.98 | 9 700 chan | nel0.0 sched switch | nrey commelting-sessiond | prev tid=1426 prev prio=20 prev state=1024 pert | omm=lttng next tid=1464 ne | ext prio=20 | | | | | * |
| BE Control 22 | fit Histogram 52 | Properties | Ell Bookmarks | | | | | | | | | N. M |
| 2 10 10 10 10 10 10 10 10 10 10 10 10 10 | in integration (23 | | ap booning | | | | | | | | | |
| | | | 3 | | | | | | | | | |
| | Selection Start 17:4 | 0:40.471 019 | 700 | | 1 I I I I I I I I I I I I I I I I I I I | | 1.1 | | | | | i |
| | Selection End 17:4 | 0:40.471 019 | 700 | | | | | | | | | |
| | Window Span | 000 100 000 | 000 | | | | | | | | | |
| | trindow span | 000.200 000 | | | | | | | | | | |
| | | | 17:40:40.469 | 95 900 | | | | | | | | 17:40:40.569 295 900 |
| | 25 | | | | | | | | | | - | |
| | | | | | | | | | | | | |
| | + | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | 124042122202 |
| | 1/wuwu.469 295 900 | | | | | | | | | | | 1/90/52/527 970 460 |
| | auto-20150518-1 | 143939/kerne | | | | | | | | | | |
| | | | | | | | | | | | | |

Figure 4. LTTng Perspective

How to Reach Us:

Home Page: nxp.com

Web Support: nxp.com/support Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including " typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, Freescale, the Freescale logo, and QorlQ are trademarks of are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2017-2018 NXP B.V.

Document Number AN5172 Revision 11.3.2, 07/2018



