LAUTERBACH
DEVELOPMENT TOOLS

Bootloader Awareness Manual coreboot

Release 09.2024

MANUAL

Bootloader Awareness Manual coreboot

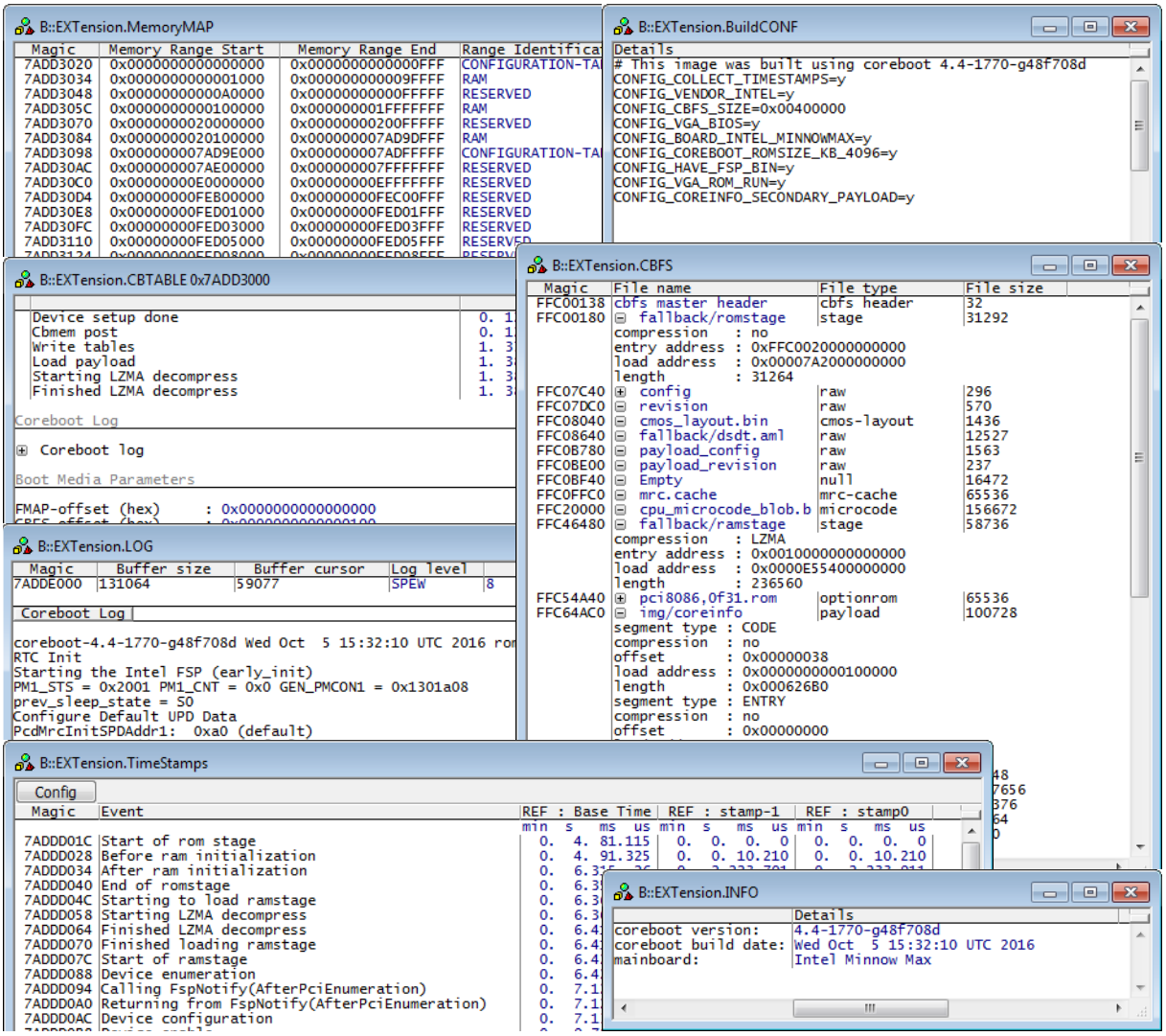
Bootloader Awareness Manual coreboot

Version 05-Oct-2024

History

28-Aug-18	The title of the manual was changed from “Bootloader Debugger for *<x>*” to “Bootloader Awareness Manual *<x>*”.

Overview



The Bootloader Awareness for coreboot contains special extensions to the TRACE32 Debugger. This chapter describes the additional features, such as additional commands and debugging approaches.

Brief Overview of Documents for New Users

Architecture-independent information:

- **"Training Basic Debugging"** (training_debugger.pdf): Get familiar with the basic features of a TRACE32 debugger.
- **"T32Start"** (app_t32start.pdf): T32Start assists you in starting TRACE32 PowerView instances for different configurations of the debugger. T32Start is only available for Windows.
- **"General Commands"** (general_ref_*x*.pdf): Alphabetic list of debug commands.

Architecture-specific information:

- **"Processor Architecture Manuals"**: These manuals describe commands that are specific for the processor architecture supported by your Debug Cable. To access the manual for your processor architecture, proceed as follows:

  - Choose **Help** menu > **Processor Architecture Manual**.

- **"OS Awareness Manuals"** (rtos_<*os*>.pdf): TRACE32 PowerView can be extended for operating system-aware debugging. The appropriate OS Awareness manual informs you how to enable the OS-aware debugging.

- **"UEFI Awareness Manuals"** (uefi_<*x*>.pdf): TRACE32 PowerView can be extended for UEFI-aware debugging. The appropriate UEFI manual informs you how to enable the UEFI-aware debugging.

Supported Versions

Currently coreboot is supported for the following versions:

- coreboot on x86 and x64 architectures

Configuration

The Bootloader Awareness for coreboot is configured by loading an extension definition file called “coreboot.t32” from the demo directory with the **EXTension.CONFIG** command. Additionally, load the “coreboot.men” menu file (see “**Coreboot specific Menu**”).

x86 32-Bit

A full configuration for x86 **32-bit** can look like this (the path prefix ~~ expands to the system directory of TRACE32):

```
; Load the debug symbols
Data.LOAD.Elf <path_to_bootblock.elf>/bootblock.elf /NoCODE
Data.LOAD.Elf <path_to_romstage.elf>/romstage.elf /NoCODE /NoClear
Data.LOAD.Elf <path_to_ramstage.elf>/ramstage.elf /NoCODE /NoClear

; Load the Bootloader Awareness for coreboot:
EXTension.CONFIG ~~/demo/x86/bootloader/coreboot/coreboot.t32

; Load the additional menu:
MENU.ReProgram ~~/demo/x86/bootloader/coreboot/coreboot.men
```

x64 64-Bit

A full configuration for x64 **64-bit** can look like this (the path prefix ~~ expands to the system directory of TRACE32):

```
; Load the debug symbols
Data.LOAD.Elf <path_to_bootblock.elf>/bootblock.elf /NoCODE
Data.LOAD.Elf <path_to_romstage.elf>/romstage.elf /NoCODE /NoClear
Data.LOAD.Elf <path_to_ramstage.elf>/ramstage.elf /NoCODE /NoClear

; Load the Bootloader Awareness for coreboot:
EXTension.CONFIG ~~/demo/x64/bootloader/coreboot/coreboot.t32

; Load the additional menu:
MENU.ReProgram ~~/demo/x64/bootloader/coreboot/coreboot.men
```

Features

The Bootloader Awareness supports the following features.

Display of coreboot Resources

The extension defines new commands to display various resources. Information on the following coreboot components can be displayed:

EXTension.INFO	Build information
EXTension.LOG	Log buffer
EXTension.CBFS	CBFS “coreboot file system”
EXTension.BuildCONF	Coreboot build configuration
EXTension.TimeStamp	Timestamps
EXTension.MemoryMAP	Memory mapping
EXTension.CBTABLE <address>	Coreboot tables content

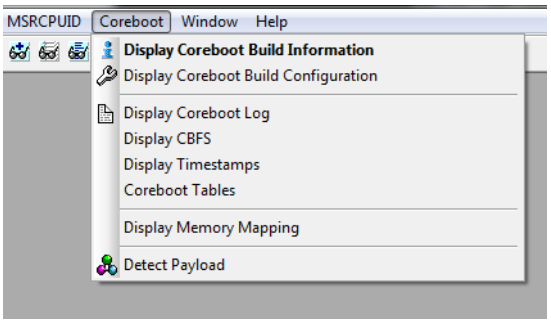
For a detailed description of each command, refer to chapter “**Coreboot Commands**”.

Since the x86/x64/Atom architecture does not allow to read memory while the program execution is running, the information can only be displayed if the program execution is stopped.

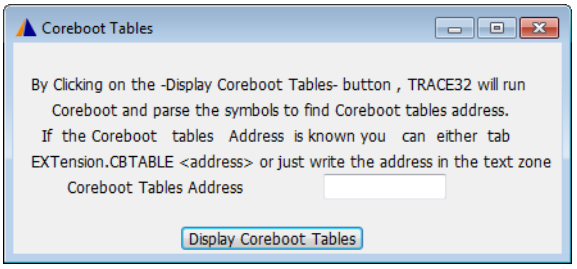
Coreboot Specific Menu

The menu file “coreboot.men” contains a menu with coreboot specific menu items. Load this menu with the **MENU.ReProgram** command.

You will find a new menu called **Coreboot**.

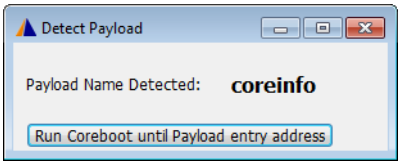


- The **Display** menu items launch the coreboot resource display windows.
- Selecting **Coreboot Tables** will open the following dialog.



The coreboot tables are created at the end of “ramstage”. The object of this dialog is to run coreboot until the coreboot tables are created and its address is set.

- Selecting **Detect Payload** will open the following dialog.

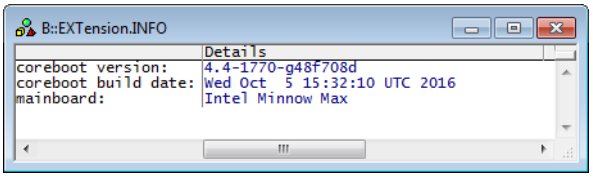


The payload name is extracted from the configuration file within the CBFS. If the payload name is not listed in the configuration file, the dialog will show an empty string.

EXTension.INFO — Display coreboot build information

Format: **EXTension.INFO**

Displays the coreboot build information (coreboot version, coreboot build date, mainboard identification).

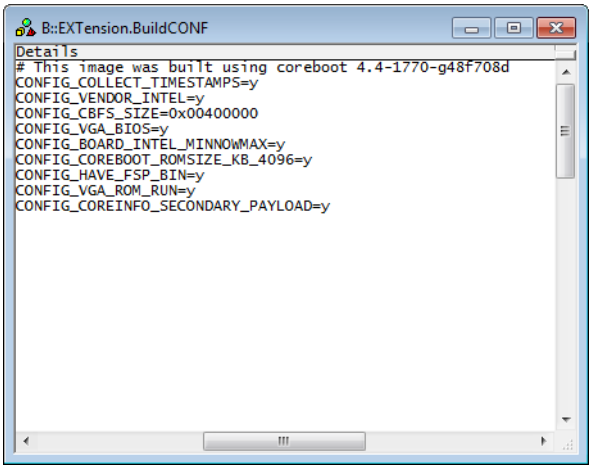


EXTension.BuildCONF — Display coreboot build configuration

Format: **EXTension.BuildCONF**

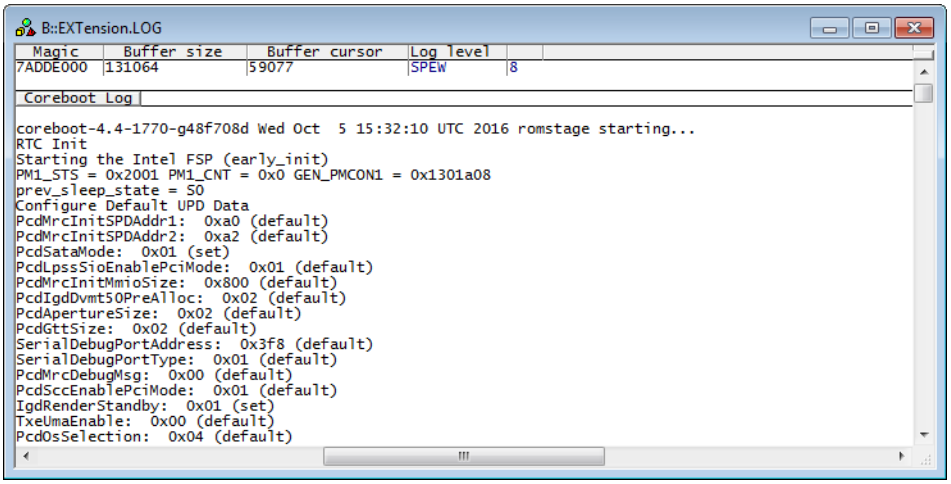
Displays the build configuration.

The configuration is extracted from the CBFS. Per default, coreboot adds the configuration file (.config) within the CBFS.



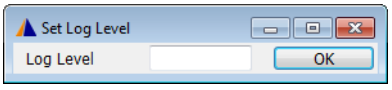
Format:	**EXTension.LOG**

The log is saved on memory in a buffer. The **EXTension.LOG** window shows the buffer size, the buffer cursor and log level.



“magic” is a unique ID, used by the Bootloader Awareness to identify the buffer address. The “magic” field is mouse sensitive. Right-clicking on it will show a local menu.

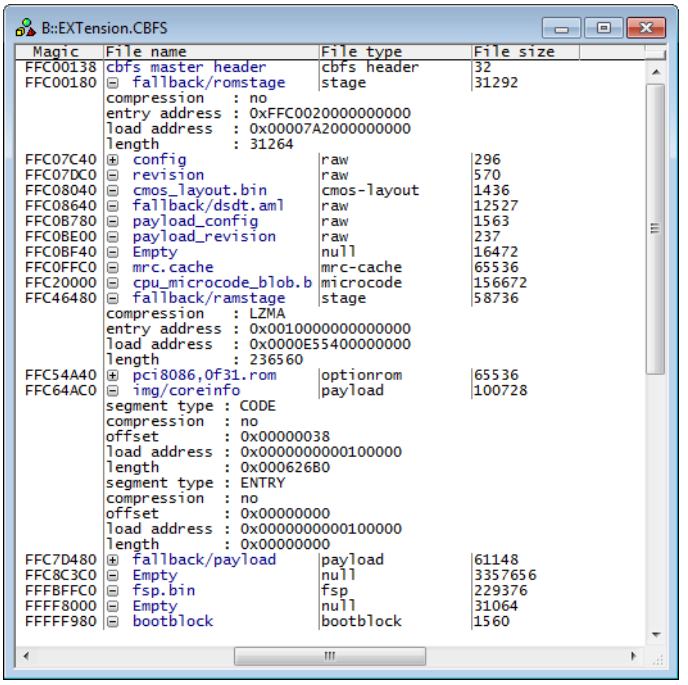
The “log level” field is mouse sensitive. Right-clicking on it will show the following dialog.



The log level can be changed during coreboot execution although it was fixed in the coreboot configuration file. Possible values are: 0 to 8.

Format:	**EXTension.CBFS**

Displays the CBFS with details (File name, type, size, and specific characteristics).



B::EXTension.CBFS

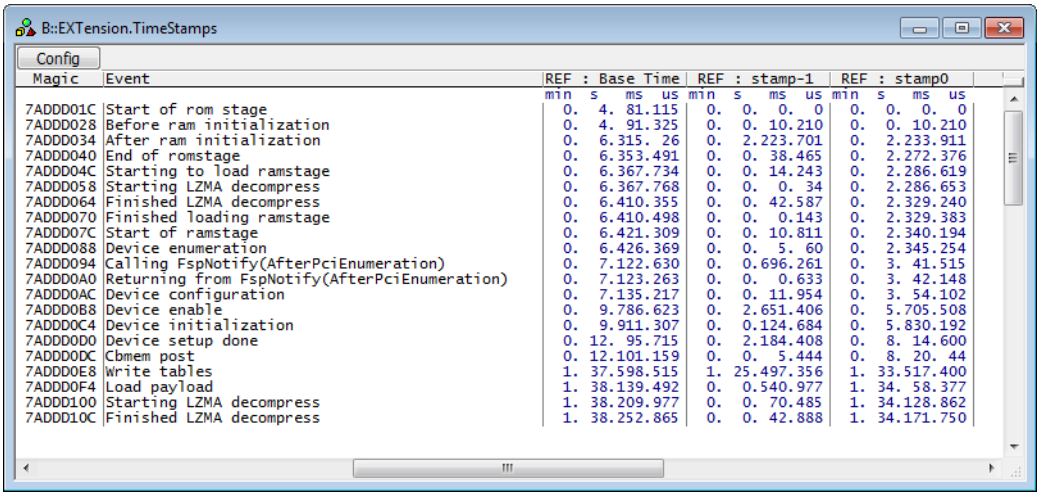
Magic	File name	File type	File size
FFC00138	cbfs master header	cbfs header	32
FFC00180	⊟ fallback/romstage	stage	31292
	compression : no		
	entry address : 0xFFC00200000000000		
	load address : 0x00007A2000000000		
	length : 31264		
FFC07C40	⊞ config	raw	296
FFC07DC0	⊟ revision	raw	570
FFC08040	⊟ cmos_layout.bin	cmos-layout	1436
FFC08640	⊟ fallback/dsdt.aml	raw	12527
FFC0B780	⊟ payload_config	raw	1563
FFC0BE00	⊟ payload_revision	raw	237
FFC0BF40	⊟ Empty	null	16472
FFC0FFC0	⊟ mrc.cache	mrc-cache	65536
FFC20000	⊟ cpu_microcode_blob.b	microcode	156672
FFC46480	⊟ fallback/ramstage	stage	58736
	compression : LZMA		
	entry address : 0x0010000000000000		
	load address : 0x0000E55400000000		
	length : 236560		
FFC54A40	⊞ pci8086,0f31.rom	optionrom	65536
FFC64AC0	⊟ img/coreinfo	payload	100728
	segment type : CODE		
	compression : no		
	offset : 0x00000038		
	load address : 0x0000000000100000		
	length : 0x000626B0		
	segment type : ENTRY		
	compression : no		
	offset : 0x00000000		
	load address : 0x0000000000100000		
	length : 0x00000000		
FFC7D480	⊞ fallback/payload	payload	61148
FFC8C3C0	⊟ Empty	null	3357656
FFFBFFC0	⊟ fsp.bin	fsp	229376
FFFF8000	⊟ Empty	null	31064
FFFFF980	⊟ bootblock	bootblock	1560

"magic" is a unique ID, used by the Bootloader Awareness to identify the address of each "file" in memory.

The "magic" fields are mouse sensitive. Right-clicking on them will show a local menu.

Format:	**EXTension.TimeStamps**

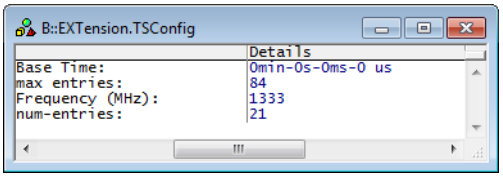
Displays all events saved by coreboot and the duration of each event.



Magic	Event	REF : Base Time	REF : stamp-1	REF : stamp0
		min s ms us	min s ms us	min s ms us
7ADDD01C	Start of rom stage	0. 4. 81.115	0. 0. 0. 0	0. 0. 0. 0
7ADDD028	Before ram initialization	0. 4. 91.325	0. 0. 10.210	0. 0. 10.210
7ADDD034	After ram initialization	0. 6.315. 26	0. 2.223.701	0. 2.233.911
7ADDD040	End of romstage	0. 6.353.491	0. 0. 38.465	0. 2.272.376
7ADDD04C	Starting to load ramstage	0. 6.367.734	0. 0. 14.243	0. 2.286.619
7ADDD058	Starting LZMA decompress	0. 6.367.768	0. 0. 0. 34	0. 2.286.653
7ADDD064	Finished LZMA decompress	0. 6.410.355	0. 0. 42.587	0. 2.329.240
7ADDD070	Finished loading ramstage	0. 6.410.498	0. 0. 0.143	0. 2.329.383
7ADDD07C	Start of ramstage	0. 6.421.309	0. 0. 10.811	0. 2.340.194
7ADDD088	Device enumeration	0. 6.426.369	0. 0. 5. 60	0. 2.345.254
7ADDD094	Calling FspNotify(AfterPciEnumeration)	0. 7.122.630	0. 0.696.261	0. 3. 41.515
7ADDD0A0	Returning from FspNotify(AfterPciEnumeration)	0. 7.123.263	0. 0. 0.633	0. 3. 42.148
7ADDD0AC	Device configuration	0. 7.135.217	0. 0. 11.954	0. 3. 54.102
7ADDD0B8	Device enable	0. 9.786.623	0. 2.651.406	0. 5.705.508
7ADDD0C4	Device initialization	0. 9.911.307	0. 0.124.684	0. 5.830.192
7ADDD0D0	Device setup done	0. 12. 95.715	0. 2.184.408	0. 8. 14.600
7ADDD0DC	Cbmem post	0. 12.101.159	0. 0. 5.444	0. 8. 20. 44
7ADDD0E8	Write tables	1. 37.598.515	1. 25.497.356	1. 33.517.400
7ADDD0F4	Load payload	1. 38.139.492	0. 0.540.977	1. 34. 58.377
7ADDD100	Starting LZMA decompress	1. 38.209.977	0. 0. 70.485	1. 34.128.862
7ADDD10C	Finished LZMA decompress	1. 38.252.865	0. 0. 42.888	1. 34.171.750

The “magic” fields are mouse sensitive. Right-clicking on them will show a local menu.

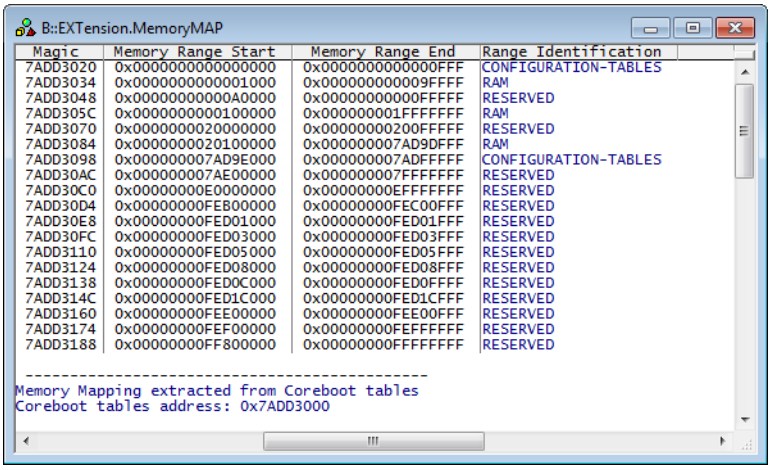
Clicking the **Config** button will display the timestamp configuration.



	Details
Base Time:	0min-0s-0ms-0 us
max entries:	84
Frequency (MHz):	1333
num-entries:	21

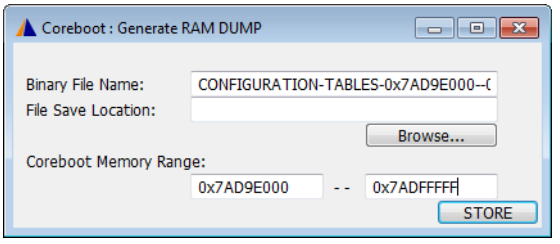
Format:	**EXTension.MemoryMAP**

Displays the memory mapping: start and end address and identification of each memory region.



The “magic” fields are mouse sensitive. Right-clicking on them will show a local menu.

A RAM dump of a specific region can be saved by selecting **generate RAM dump** from the local menu of the “magic” field. The binary name and memory range addresses will be set automatically.



Format: **EXTension.CBTABLE** *<address>*

Displays the contents of coreboot tables.

<address> Coreboot table address