



TF-M Logging API

Jackson Cooper-Driver
10/04/25

Aims of the new library

- Simplify logging in TF-M
- Unify macros across components
- Simplify configuration
- Reduce footprint compared to printf

Logging API

- Privileged and unprivileged API
 - INFO, WARN, ERROR, etc.
 - Logs directly to UART
 - Used by bootloaders, SPM, PSA-ROT partitions
 - INFO_UNPRIV, WARN_UNPRIV, ERROR_UNPRIV, etc.
 - Logs via SVC
 - Used by APP-ROT partitions
- Different logging levels (ERROR, INFO, etc.)
 - By default, prints a prefix with the string ([ERR], [INF], etc.)
 - _RAW macros available which do not output the prefix
- Level to output controlled by different config variable for each component
 - TFM_SPM_LOG_LEVEL=LOG_LEVEL_NONE
 - TFM_BL1_LOG_LEVEL=LOG_LEVEL_INFO
- Standard format specifiers available
 - %u, %d, %i, %x, %s, %lx

Example usage

Privileged

```
#include "tfm_log.h"
```

```
INFO("Hello world\n")  
-> [INF] Hello World
```

```
uint32_t x = 0xAB;  
VERBOSE_RAW("My int: %x\n", x)  
-> My int: ab
```

Unprivileged

```
#include "tfm_log_unpriv.h"
```

```
INFO_UNPRIV("Hello world\n")  
-> [INF] Hello World
```

```
uint32_t x = 0xAB;  
VERBOSE_UNPRIV_RAW("My int: %x\n", x)  
-> My int: ab
```

Customization

- `set(TFM_PARTITION_LOG_LEVEL LOG_LEVEL_VERBOSE CACHE STRING "Set debug SP log level as Debug level")`
- `cmake -S ... -B ... -DTFM_BL1_LOG_LEVEL=LOG_LEVEL_NONE ...`
- ```
#undef LOG_LEVEL
#define LOG_LEVEL LOG_LEVEL_VERBOSE
#include "tfm_log.h

VERBOSE(...);
```