

# DLT usage guide and examples



The purpose of this document is to give some more "hands on" examples, especially in the context of how to do good logging. For Basic usage examples of DLT please refer to links below

## Cheatsheet

- [DLT cheatsheet](#)

## Usage of DLT

### Cosmetic DLT\_STRING

The DLT module is providing some macros to log data. Please don't add things like "\n", "\r", " " (spaces) e.c.

### DLT\_RAW usage

Avoid using DLT\_RAW for data that is less or equal to 4 bytes in size. Each time DLT\_RAW is used, 2 bytes for the length information is added.

The DLT RAW Frame is constructed like this:

DLT_RAW	Info	Data_Length	Data
	Length	2	XX

#### Suboptimal solution:

```
DLT_LOG_ID7(CH_CONTEXT_STATE, DLT_LOG_DEBUG, 59147,
    DLT_CSTRING("ExitOnError: ("),
    DLTRAW((void*)&SELF(sink)->flbockId, 1),
    DLT_CSTRING(", "),
    DLT_RAW((void*)&SELF(sink)->insId, 1),
    DLT_CSTRING(", "),
    DLT_RAW((void*)&SELF(sink)->sinkNr, 1),
    DLT_CSTRING(")"));
```

This log output will effectively send:

- Header of 20 bytes
- 9 bytes (3 bytes per DLT RAW)

Total 29 header bytes

#### A solution like this is better

```
DLT_LOG_ID7(CH_CONTEXT_STATE, DLT_LOG_DEBUG, 59147,
    DLT_CSTRING("ExitOnError: ("),
    DLT_UINT8(SELF(sink)->flbockId),
    DLT_CSTRING(", "),
    DLT_UINT8(SELF(sink)->insId),
    DLT_CSTRING(", "),
    DLT_UINT8(SELF(sink)->sinkNr),
    DLT_CSTRING(")"));
```

This log output will effectively send:

- Header of 20 bytes
- 3 bytes (DLT\_UINT8)

Total 23 header bytes

We freed 6 bytes in one message by using more effective types!

## Group DLT log's

Each DLT message has a header which consumes 20 bytes. There for grouping related information can save a lot of resources. Another advantage of grouping necessary information is that if related information is split into multiple messages these messages are not necessarily printed after each other because they can be interrupted by messages of other processes. Please also refer to [Combine multiple messages in the trace guideline](#)

A bad example:

```
DLT_LOG(mycontext1,DLT_LOG_INFO,DLT_CSTRING("Total frames: "), DLT_UINT16(1000));
DLT_LOG(mycontext1,DLT_LOG_INFO,DLT_CSTRING("Sync frames: "), DLT_UINT8(0));
DLT_LOG(mycontext1,DLT_LOG_INFO,DLT_CSTRING("Reem frames: "), DLT_UINT8(0));
DLT_LOG(mycontext1,DLT_LOG_INFO,DLT_CSTRING("Valid frames: "), DLT_UINT16(100));
DLT_LOG(mycontext1,DLT_LOG_INFO,DLT_CSTRING("Urgent frames: "), DLT_UINT8(0));
```

Output:

```
Total frames: 1000
Sync frames: 0
Reem frames: 0
Valid frames: 100
Urgent frames: 0
```

Better to aggregate information like this:

```
DLT_LOG(mycontext1,DLT_LOG_INFO, DLT_CSTRING("Frame info: ,"),
        DLT_CSTRING("total="),DLT_UINT16(1000),DLT_CSTRING(", ")
        DLT_CSTRING("sync="),DLT_UINT8(0),DLT_CSTRING(", ")
        DLT_CSTRING("reem="),DLT_UINT8(0),DLT_CSTRING(", ")
        DLT_CSTRING("valid="),DLT_UINT16(100),DLT_CSTRING(", ")
        DLT_CSTRING("urgent="),DLT_UINT8(1))
```

Output:

```
Frame info: total=1000, sync=0, reem=100, valid=0, urgent=1
```

In this example 4\*20 bytes are just saved because of the header information. Additionally this information is much easier to analyze.

## Log structures recommendation

### Structuring conditional parts

When you have to log results of conditional cases avoid to add a log just before or just after the conditional part. Merge logs as well as possible.

#### Suboptimal solution

```
if (I_res==TRUE)
{
    DLT_LOG(mycontext1,DLT_LOG_INFO, DLT_CSTRING("Verify ABC Signature: Signature Check result ok"));
}
else
{
    DLT_LOG(mycontext1,DLT_LOG_INFO, DLT_CSTRING("Verify ABC Signature: Signature Check result ERROR!"));
}
DLT_LOG(mycontext1,DLT_LOG_INFO, DLT_CSTRING("Result code of ABC Siganture verification: "),
        DLT_INT(parameter_ptr->result));
```

#### Better solution

```

if (I_res==TRUE)
{
    DLT_LOG(mycontext1,DLT_LOG_INFO, DLT_CSTRING("Verify ABC Signature: Signature Check result ok, result
code: "),
            DLT_INT(parameter_ptr->result));
}
else
{
    DLT_LOG(mycontext1,DLT_LOG_ERROR, DLT_CSTRING("Verify ABC Signature: Signature Check result ERROR, result
code: "),
            DLT_INT(parameter_ptr->result));
}

```

We gained 20 bytes from the header and all information is compactly available in one point.

## Logging a switch statement

(in a loop)

### suboptimal solution

#### Suboptimal solution

```

for(i = 0; i < VALUE_4, i++)
{
    switch(i)
    {
        case VALUE_0:
            Base = Value_0;
            /*Do something*/
            DLT_LOG(mycontext9,DLT_LOG_DEBUG,DLT_CSTRING("Checked value 0"));
            break;
        case VALUE_1:
            Base = Value_1;
            /*Do something*/
            DLT_LOG(mycontext9,DLT_LOG_DEBUG,DLT_CSTRING("Checked value 1"));
            break;
        case VALUE_2:
            Base = Value_2;
            /*Do something*/
            DLT_LOG(mycontext9,DLT_LOG_DEBUG,DLT_CSTRING("Checked value 2"));
            break;
        case VALUE_3:
            Base = Value_3;
            /*Do something*/
            DLT_LOG(mycontext9,DLT_LOG_DEBUG,DLT_CSTRING("Checked value 3"));
            break;
    }
}

```

### optimal solution

```
for(i = 0; i < VALUE_4, i++)
{
    switch(i)
    {
        case VALUE_0:
            Base = Value_0;
            /*Do something*/
            break;
        case VALUE_1:
            Base = Value_1;
            /*Do something*/
            break;
        case VALUE_2:
            Base = Value_2;
            /*Do something*/
            break;
        case VALUE_3:
            Base = Value_3;
            /*Do something*/
            break;
    }
}
DLT_LOG(mycontext9,DLT_LOG_DEBUG,DLT_CSTRING("Checked values 0-3"));
```

---

*Transferred from MediaWiki. Original author and history:  
Created: 13:36, 13 January 2014, Birk.bremer  
Last edit **before** transfer: 07:34, 13 April 2015, Jeremiah  
[for Latest changes -> View Page History in this Wiki](#)*