

# Unified logger

## Simple logger library used to unify logging pattern.

Jednoduchá knižnica pre zosynchronizovanie stdout, a riešenie “string interleaving” vo viacvláknovom prostredí.

`unified_logger.cpp`

`unified_logger.h`

`locked_stream.cpp`

`locked_stream.h`

`Makefile`

`unified_logger_demo.cpp`

## Spustenie

Zdrojové súbory vo vetve `unified_logger` a zložke `unified_logger`. Pre spustenie demo aplikácie existuje pravidlo:

```
make run
```

## Použitie

Príklad použitia je v demo aplikácii v `unified_logger_demo.cpp`.

Pre logovanie sa využíva globálny objekt `Unified_logger`, ktorý podporuje stream ooperátor `<<` pre zapisovanie správ

Vytvorenie loggeru spočíva z názvu loggeru, zložky pre logovacie súbory a minimálnej úrovne pre správy.

```
Unified_logger logger("framework", ".", LogSeverity::INFO);
```

Parametry pre logovaciu zložku (“.”) a log level (`LogSeverity::WARN`) sú nepovinné.

Minimálna konfigurácia pozostáva z názvu loggeru:

```
Unified_logger logger("framework");
```

Nepovinné parametry možno manuálne nastaviť pomocou dvojice funkcií:

```
void setLogLevel(LogSeverity log_level);
```

```
void setLogFolderPath(std::string folder_path);
```

Pre logovanie na stdout možno použiť funkciu apomocné makro, ktoré automaticky dopĺňa prvé dva parametre funkcie:

```
logger.out(FILE, LINE, "manager", "TRACE") << FUNCTION << std::endl;
```

```
logger.LOGOUT("temperature", "DEBUG") << "Entered config state" << std::endl;
```

Obdobné funkcie existujú pre logovanie do súboru:

```
logger.file(FILE, LINE, "manager", "TRACE") << FUNCTION << std::endl;
```

```
logger.LOGFILE("manager", "INFO") << "Manager initialized" << std::endl;
```

## Log format:

[dd.mm.yyyy hh:mm:ss:xxx] tag [thread\_id] origin\_file:origin\_line severity message

## Datetime

podľa normy ISO 8601

## Tag

Hierarchický formát identifikácie zdroju správy

beeeon.hostname.logger\_id.tag

beeeon.ant-2.framework2.manager

## Thread\_id

podľa funkcie `std::this_thread::get_id()`

14miestny číselný identifikátor

## File:Line

poloha v zdrojovom súbore kde správa vznikla

\_\_FILE\_\_, \_\_LINE\_\_

automaticky vkladané pri použití makier LOGOUT, LOGFILE

## Severity

enum v unified\_logger.h

ALL - pre nastavenie loggeru na zachytávanie všetkých správ

TRACE - veľmi podrobné informácie, výpis volania funkcií, výpis staovvých premenných

MSG - výpis príchodzej/odchodzej komunikácie

DEBUG - jemnejšie debugovacie informácie o stave

WARN - nesúlad v konfigurácií, chyby ktoré nemajú vplyv na beh aplikácie

INFO - dôležité informácie z prijatých správ určené na hrubú lokalizáciu chýb

ERROR - chyby z ktorých sa aplikácia môže zotaviť

FATAL - chyby ktoré vedú k zlyhaniu aplikácie

NONE - pre nastavenie loggeru na zahadzovanie všetkých správ

## Message

Ľubovolná užívateľská správa

```
(2016-04-07 13:58:08:765) humidity [47984377546496] unified_logger_demo.cpp:8 INFO Pa  
rameters: unit id: 124 status: 42 message: accepted
```

```
(2016-04-07 13:58:08:765) temperature [47984377546496] unified_logger_demo.cpp:9 DEBU  
G Entered config state
```

```
(2016-04-07 13:58:08:765) pressure [47984377546496] unified_logger_demo.cpp:10 FATAL  
Incorrect config: ID not positive value: 0
```

```
(2016-04-07 13:58:08:766) manager [47984379647744] unified_logger_demo.cpp:15 INFO Ma  
nager initialized
```

Log file format:

zakódovaný názov loggeru a aktuálny dátum

TAGyyyymmdd.log