Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

EiffelRSS

LOGFILE: Developer Guide

Michael Käser <kaeserm@student.ethz.ch>
Martin Luder <luderm@student.ethz.ch>
Thomas Weibel <weibelt@student.ethz.ch>

Informatik
Computer Science

Abstract

`LOGFILE` represents a file which can be used for logging messages during the program execution.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

LOGFILE represents a file which can be used for logging messages during the program execution.

Each message has its own priority (a positive integer value) and each logfile a certain user defined threshold. If the priority of the message is greater or equal than the threshold, it gets written to the log file together with a timestamp.

See figure 1.1 for an overview of the class.

1.2 Usage

```
class USAGE_EXAMPLE

create make

feature -- Initialization

  make is
      -- Creation procedure.
    do
      -- Create logfile
      create log_file.make_filename("logfile")

      -- Set the threshold
      log_file.set_threshold(log_file.Error)

      -- Output a message with one line and priority critical
      log_file.log_message("One line test message", log_file.Critical)
```

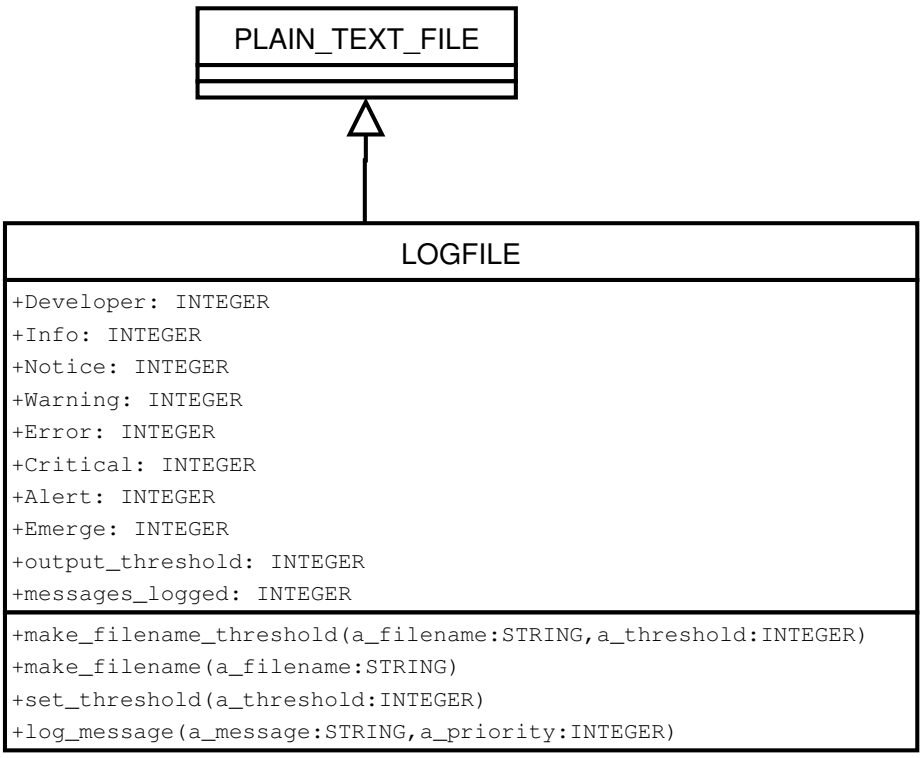

Figure 1.1: UML diagram of `LOGFILE`

```
        -- Output a message which won't be logged
        log_file.log_message("Not important enough", ↘
        →log_file.Info)

        -- Output a multi-line message
        log_file.log_message("Line One%NLine Two%NLine ↘
        →Three", log_file.Error)
    end

feature -- Arguments

  log_file: LOGFILE
      -- Logfile

end -- class USAGE_EXAMPLE
```

The output of the above example is

```
[16:02:37 05 JAN 2005]: One line test message
[16:02:37 05 JAN 2005]: Line One
                        Line Two
                        Line Three
```

Chapter 2

Features

2.1 Initialization

2.1.1 make_filename

```
make_filename (a_filename: STRING) is
  -- Create logfile object with a_filename as file name
```

Note that the default threshold which will be set if you create the object with this procedure is zero.

2.1.2 make_filename_threshold

```
make_filename_threshold (a_filename: STRING; a_threshold
→: INTEGER) is
  -- Create logfile object with a_filename as file name
  →and a_threshold as output threshold
```

2.2 Access

2.2.1 messages_logged

```
messages_logged: INTEGER
  -- The number of messages which got logged
```

2.2.2 output_threshold

```
output_threshold: INTEGER
  -- The current threshold used
```

2.3 Constants

```
Developer, Info, Notice, Warning, Error, Critical, Alert
→, Emerge: INTEGER is unique
  -- Some predefined constants which can be used as a
  →priority
```

2.4 Basic Operations

2.4.1 set_threshold

```
set_threshold (a_threshold: INTEGER) is
  -- Set the output threshold to a_threshold
```

Sets a new threshold, which must be a positive integer value. Each message which the object receives after the execution of this command is only written to the logfile if its priority is greater or equal the threshold.

2.4.2 log_message

```
log_message (a_message: STRING; a_priority: INTEGER) is
  -- Log the message to the logfile if a_priority is
  →equal or greater than the threshold
```

This is the actual command to send a message to the object. If the priority is high enough, a timestamp with format

```
  [[0]hh:[0]mi:[0]ss [0]dd mmm yyyy]:
```

gets added. If the message containts line breaks, all lines will get correctly indented.

Attention: Note that there is no guarantee that the operating system will physically write the data to the disk. At least it will end up in the buffer cache, making the data visible to other processes.