EffortLog v1.0.0

Contents

Using the Help Browser

You can get help by hitting **help** from the application menu bar under Help. Additionally you can always bring up this help browser by hitting **F1** from your keyboard if you are running Windows or Linux operating systems or **Command+?** if you are on Mac OS X.

This help is split into multiple files which are connected through links. These are highlighted with a blue color of the text. Additionally each of the help files provides a simple navigation in the upper part of the window. This navigation will guide you through the whole help document if you hit the **next** button. You can go back to the last help file with the **back** button. Additionally you can go at any time back to the top of the documentation by hitting the **top** button.

The help browser might look like figure 1.

Configuration Wizard

The settings dialog is the first window to open upon execution of the program. Its main goal is it to present the user with a convenient tool to set up the configuration needed for monitoring programming effort.

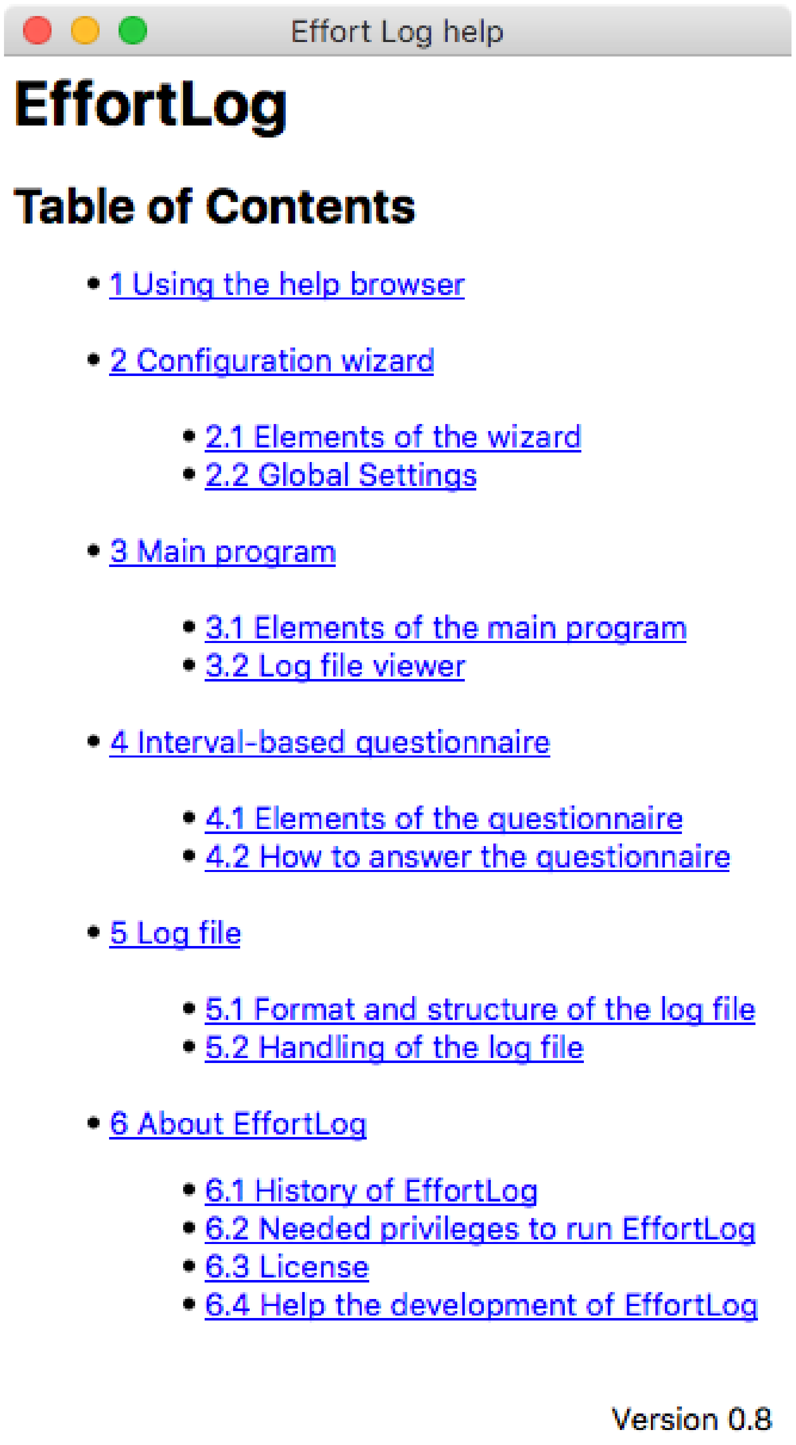
Elements of the Wizard

The configuration dialog consists of one unified window which provides access to all user-input needed. The dialog holds a vertical list of entries. Each entry is divided into a label which explains the expected input, a filed for the input and optional a button for easier access to the expected input. The user can browse through the input fields by hitting "tab" on the keyboard. The main window is divided into four distinct areas which will be explained in detail in the following section. The configuration wizard is depicted in figure 2.

Global Settings

The configuration wizard will set the following user-specified global settings of the main program:

- The specified user name to distinct between multiple users working on the same project.
- The name of the project to distinct between multiple projects a user might work on.
- The current development stage of the project.
- The logging interval in minutes which sets the default interval at which the questionnaire is executed.
- The output directory to which the log file is saved.
- The name of the output file.

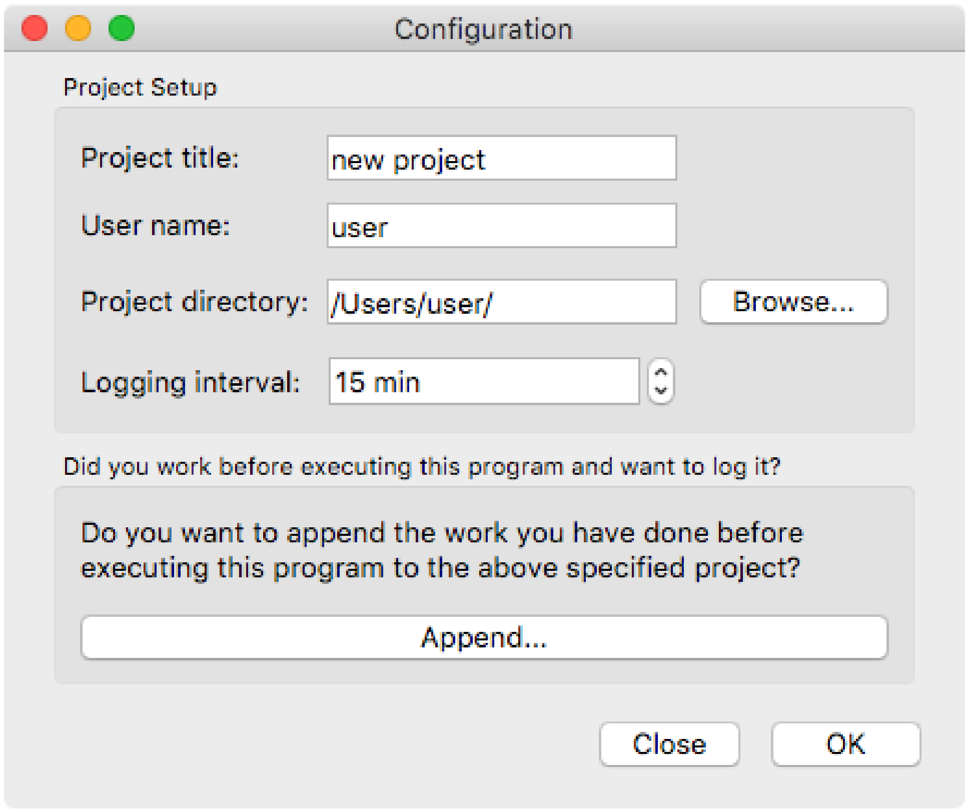


EffortLog

Table of Contents

Version 0.8

Figure 1: The help window.


Configuration

Project Setup

Project title: new project

User name: user

Project directory: /Users/user/

Logging interval: 15 min

Did you work before executing this program and want to log it?

Do you want to append the work you have done before executing this program to the above specified project?

Figure 2: The configuration wizard.

Main program

The main program of Effort Log is a simple window providing the user with information about the running logging.

Elements of the Main Window

The main window in EffortLog consists of various elements which described in detail below. Its main purpose is to maintain a simple interface to provide the user about the current project he or she is logging to and when the next scheduled logging event will be. Except of the menubar and the toolbar, there are no elements which need user interaction. The main window looks like figure 3.

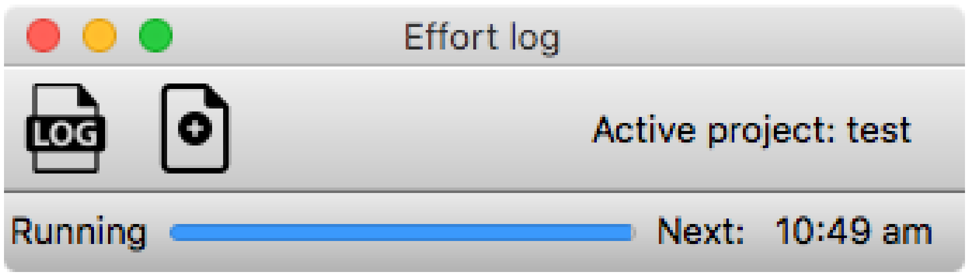


Figure 3: The main window.

Menubar

A simple menubar provides the user with information about the program.

File Log current effort

Starts a questionnaire for logging the current effort. This can be useful e.g. if the user wants to log earlier than the specified logging interval.

Read current log file

Opens the current log file for quickly reading through it.

Help About

Display information about EffortLog including the version and author of the program.

Help

Open the help window displaying the contents of the EffortLog help.

A simple toolbar provides quick access to to functions of the menubar and information about the currently active project.

Statusbar

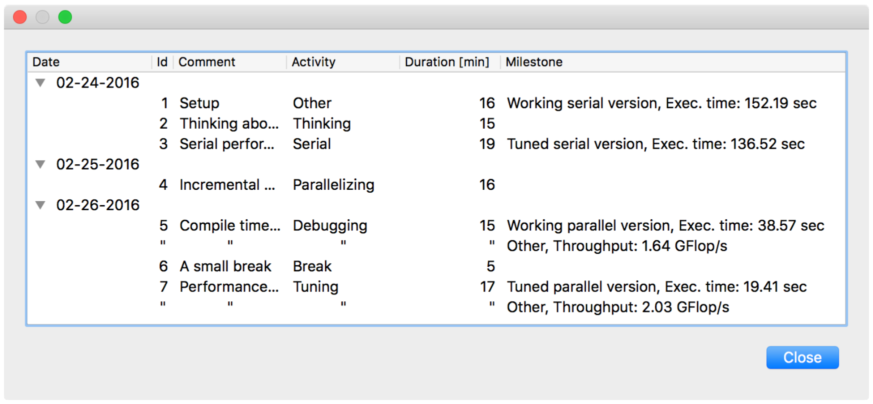
A statusbar in the lower part of the main window provides the user with information about the running program. The statusbar displays a message with the following information:

- Indicator if the program is running.
- A timer in form of a progress bar until the next event.
- The time of the next scheduled logging event.

This message can be overwritten by tooltips when hovering with the mouse pointer over a button.

Elements of the Log File Viewer

The log file viewer consists of a convenient tree view to allow for a quick overview over project's activities and milestones during the active development. It can be used to review the main development efforts and performance improvements. The activities are grouped by date and hold their identifier, comment, activity, development duration in minutes and if available the according milestone(s). If a milestone exists for the activity, the type of milestone with the respective performance measure is displayed. The respective activities of a date can be viewed by clicking on the date in the first column of the viewer and can be hidden by clicking it again. A close button is located in the lower right of the window. Note that this window holds a viewer only. No changes to the underlying log file can be made. The log file viewer is depicted in figure 4.



Date	Id	Comment	Activity	Duration [min]	Milestone
▼ 02-24-2016					
	1	Setup	Other	16	Working serial version, Exec. time: 152.19 sec
	2	Thinking abo...	Thinking	15	
	3	Serial perfor...	Serial	19	Tuned serial version, Exec. time: 136.52 sec
▼ 02-25-2016					
	4	Incremental ...	Parallelizing	16	
▼ 02-26-2016					
	5	Compile time...	Debugging	15	Working parallel version, Exec. time: 38.57 sec
	"	"	"	"	Other, Throughput: 1.64 GFlop/s
	6	A small break	Break	5	
	7	Performance...	Tuning	17	Tuned parallel version, Exec. time: 19.41 sec
	"	"	"	"	Other, Throughput: 2.03 GFlop/s

Figure 4: The log file viewer.

Interval-based Questionnaire

This questionnaire is executed in a chosen interval and asks the user about his or her current development's effort.

Elements of the Questionnaire

The window of the questionnaire consists of the following elements.

Statusbar

A simple statusbar in the upmost part of the window shows the number of successfully logged events during the current session.

Section 1: Activity

The upper part of the window shows a free-form filed to specify the activity one was working on. It features auto-completion on previous types of activities.

Section 2: Performance

The second section allows the use to specify performance-related data. If features radio buttons with "Yes" and "No" where a single selection is allowed. On specifying yes, a set of questions appear. The first one handles the performance measurements in form of a metric and the raw performance number. The choice for the metric include: execution time in secondes, minutes and hours, throughput in GFlop/s, the speedup compared to an optimized serial version of the code and other which allows for free-form input.

The second question concerns the leveraged parallelism during the measurements. The choice for the metric are number of threads or nodes and other which allows for free-form input. Next are four questions in the same style of a simple input form. The questions are regarding the used architecture, compiler, programming model and data size for the measurements. All four fields feature auto-completion on previous inputs.

Section 3: Comment

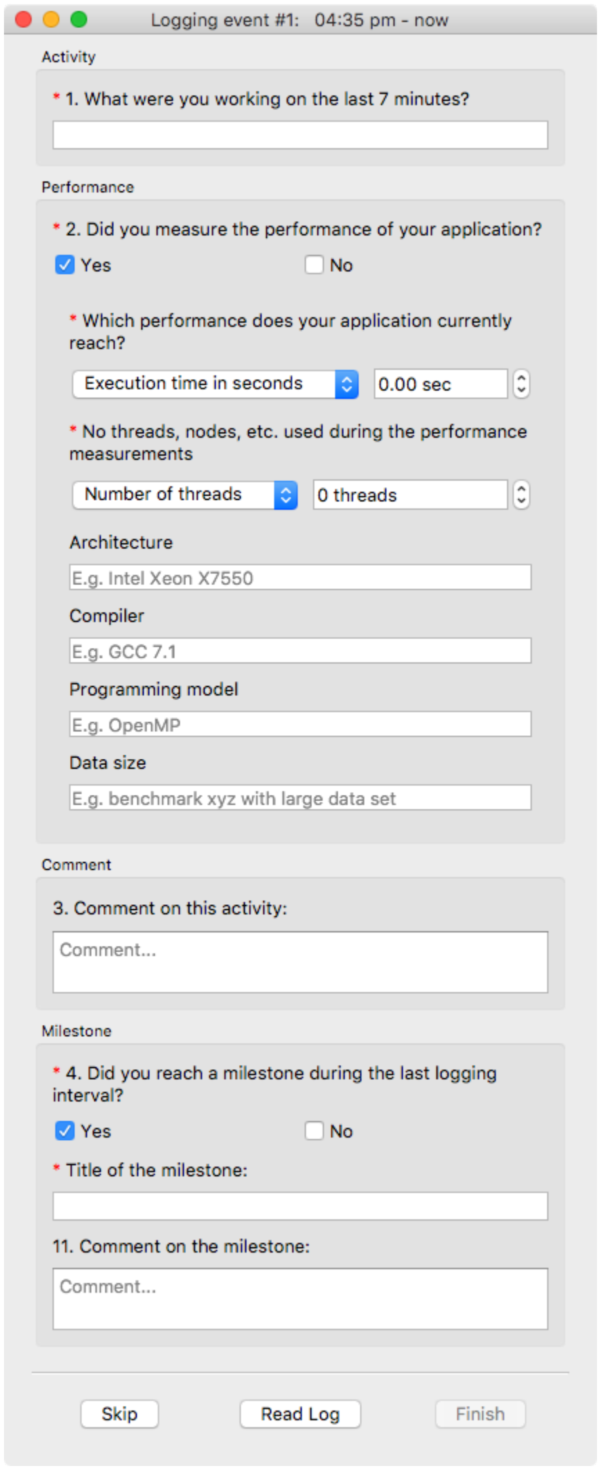
The third section holds a text field to input comments on the current activity.

Section 4: Milestone

The lower part of the window allows the use to specify milestone-related data. If features radio buttons with "Yes" and "No" where a single selection is allowed. On specifying yes, two more input forms appear. The first one is a free-form input field to specify the title of the milestone. Typical choices include: working serial version, tuned serial version, working parallel version and tuned parallel version. The second field lets one specify additional information as a comment linked to the milestone.

Buttons

The are three buttons in the lower part of the window:



Logging event #1: 04:35 pm - now

Activity

* 1. What were you working on the last 7 minutes?

Performance

* 2. Did you measure the performance of your application?

[✓ Yes] No

* Which performance does your application currently reach?

Execution time in seconds | 0.00 sec

* No threads, nodes, etc. used during the performance measurements

Number of threads | 0 threads

Architecture

E.g. Intel Xeon X7550

Compiler

E.g. GCC 7.1

Programming model

E.g. OpenMP

Data size

E.g. benchmark xyz with large data set

Comment

3. Comment on this activity:

Comment...

Milestone

* 4. Did you reach a milestone during the last logging interval?

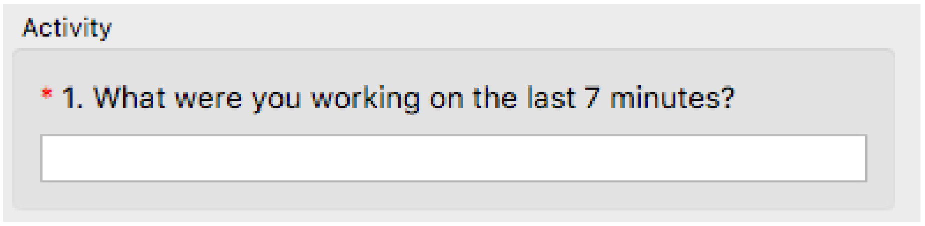
[✓ Yes] No

* Title of the milestone:

11. Comment on the milestone:

Comment...

Figure 5: The questionnaire window.


Activity

* 1. What were you working on the last 7 minutes?

Figure 6: Section 1: Activity

- Skip: If the user hits this button the current logging event will be skipped and the user is presented with the main program. The timer until the next event will be reset to the specified logging interval. Hotkey: **q**
- Read Log: Opens a view on the logging events of the current project. See Elements of the Log File Viewer for additional information on the viewer.
- Finish: If the user finished all input he or she can hit this button to save the current logging event and return to the main program. The timer until the next event will be reset to the specified logging interval. Hotkey: **f**

How to Answer the Questionnaire

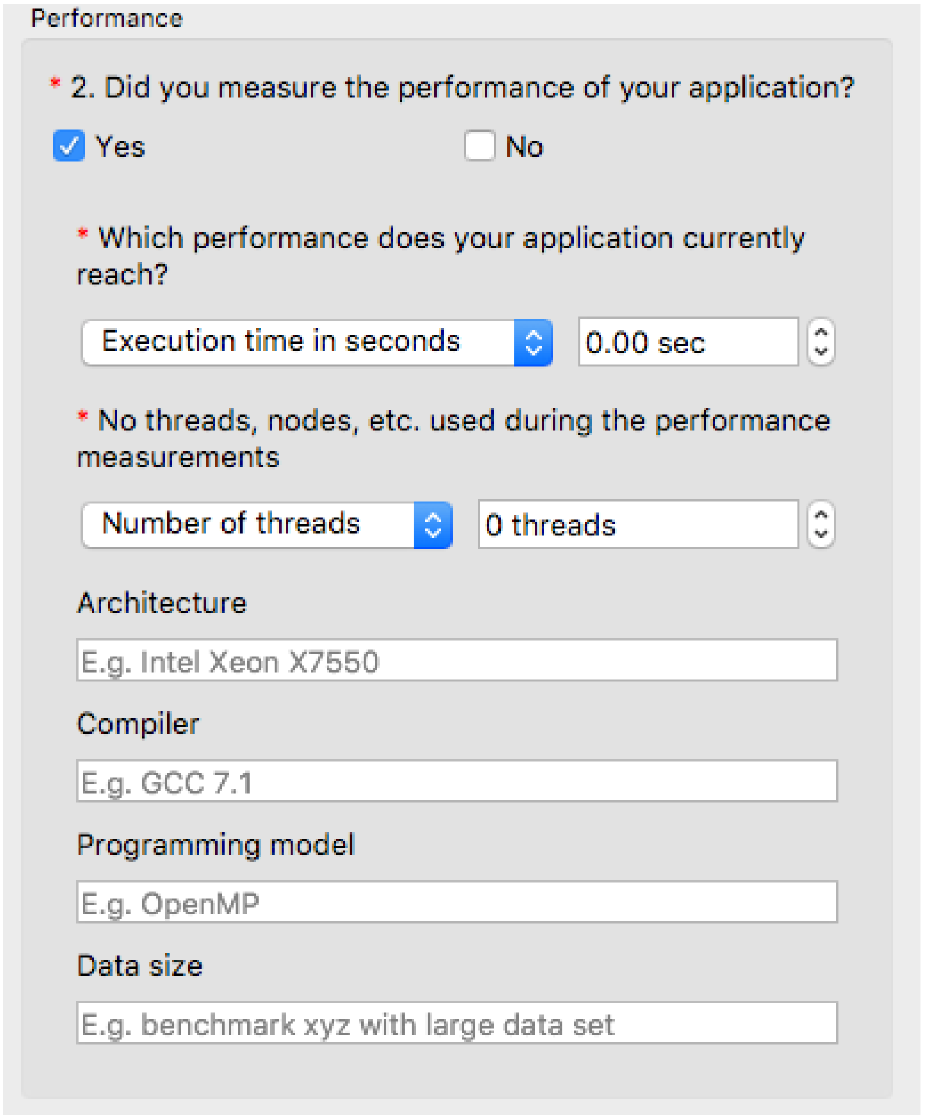
There are some default activities to choose:

- Break
- Thinking
- Serial
- Parallelizing
- Testing
- Debugging
- Tuning
- Experimenting

To get the results of the logs as accurate as possible please only choose an activity from this list if it falls under this category. If you are uncertain if it belongs to one of the categories above or you worked on something different please choose other from the list. If you did choose other activity please describe it in detail in the comment field.

The comment field is used to keep track of additionally information you want to add to your logs.

If you want to take a brake for some time longer than the default logging interval you do not have to quit the program. You might just let it run in the background and it will set a bigger time slot for your break as soon as you return back to your work place.



Performance

* 2. Did you measure the performance of your application?

[✓ Yes] No

* Which performance does your application currently reach?

Execution time in seconds: 0.00 sec

* No threads, nodes, etc. used during the performance measurements

Number of threads: 0 threads

Architecture

E.g. Intel Xeon X7550

Compiler

E.g. GCC 7.1

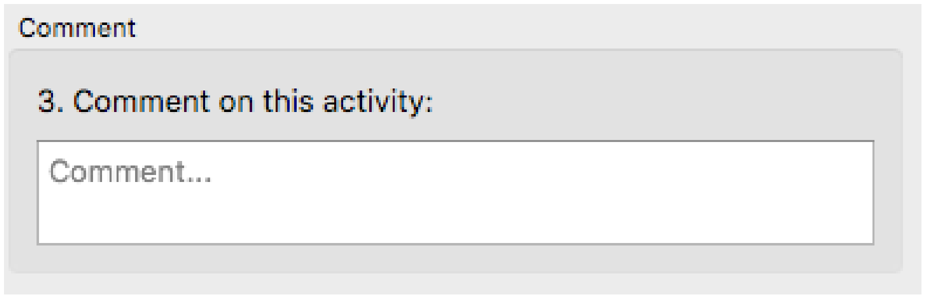
Programming model

E.g. OpenMP

Data size

E.g. benchmark xyz with large data set

Figure 7: Section 2: Performance

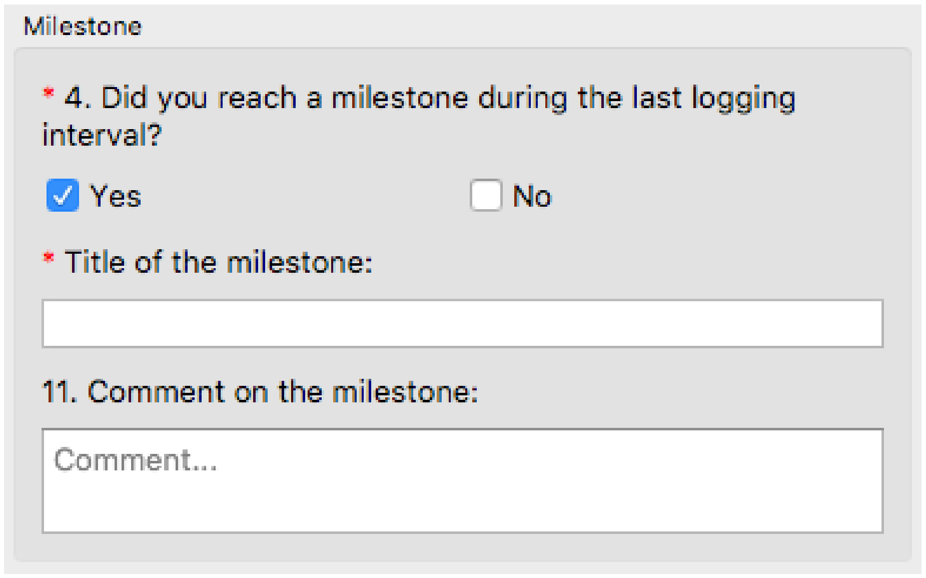


Comment

3. Comment on this activity:

Comment...

Figure 8: Section 3: Comment



Milestone

* 4. Did you reach a milestone during the last logging interval?

[✓ Yes] No

* Title of the milestone:

11. Comment on the milestone:

Comment...

Figure 9: Section 4: Milestone

Log File

This help describes the log file which holds all information collected by EffortLog. Its main purpose is to further evaluate the developer's effort.

Format and Structure of the Log File

The log file is formatted in JSON. It contains of the following entries:

- "InitialProjectStage": The initial development stage of the project. This was specified during the configuration wizard. The choices for the stage are:
  - Scratch
  - Working serial version
  - Tuned serial version
  - Working parallel version
  - Tuned parallel version
- "InitialProjectStageComment": A user-provided comment on the above development stage.
- "LoggingEvents": A list of the logging events sorted by the occurrence. Each entry consists of the following keys:
  - "ID": A unique identifier of the event. The first event holds id 0 and all following events are then consecutively numbered.
  - "UserName": The user's name. This was specified during the configuration wizard.
  - "ProjectTitle": The title of the project. This was specified during the configuration wizard.
  - "ProjectStage": The current development stage of the project. This was specified during the configuration wizard. The choices for the stage are:
    - Scratch
    - Working serial version
    - Tuned serial version
    - Working parallel version
    - Tuned parallel version
  - "ActivityType": The user-specified type (free-form) of activity. The user is provided with a list of default activities:
    - Break
    - Thinking
    - Serial
    - Parallelizing
    - Testing
    - Debugging
    - Tuning
    - Experimenting
  - "Comment": Contains a comment on the activity. If the user did not

enter any comment this field is empty.
- "CurLoggingTime": The time when the user finished the questionnaire.
- "Interval": The time in minutes since the last stored logging event.
- "LastLoggingTime": The time when the user finished the questionnaire the time before the current event.
- "LoggingInterval": The specified logging interval. This was specified during the configuration wizard.
- "NoEventsCurrentSession": A count of the number of events logged during the current session.
- "Scheduler": Specifies how the event was triggered. -1 denotes an unknown or undefined event, 0 denotes an interval-based event , 1 an appended event, 2 a manual event executed through the GUI, and 3 an event on closing the program.
- "Architecture": The architecture used during the performance measurements.
- "Compiler": The compiler used for generating the binary which was executed for the performance measurements.
- "DataSize": The problem size used during the performance measurements.
- "NoThreadsNodes": The number of threads/nodes which were utilized for the performance measurements.
- "PerfComment": The obtained performance measure.
- "PerfMetric": The performance metric used.
- "ProgrammingModel": The programming model used.
- "ThreadsNodes": Distinguishes if threads or nodes are used for the key NoThreadsNodes.
- "MsID": A unique identifier of the milestone. The first event holds id 0 and all following events are then consecutively numbered. The ID is -1 if no milestone was specified.
- "MSTitle": The user-defined type (free-form) of milestone. The default choices include:
  - Working serial version
  - Tuned serial version
  - Working parallel version
  - Tuned parallel version
- "MsComment": Contains a comment on the milestone. If the user did not enter any comment this field is empty.

Note: The JSON keys are typically alphabetically ordered by the Qt JSON library. Note 2: The tool expects unaltered JSON files. Do not change the log files manually!

Handling of the Log File

There are two different ways how EffortLog handles the log file:

- If the specified log file does not exist EffortLog will create it and start writing to it.
- If the specified log file does exist EffortLog will append to the end of file. This way, old logs are never overwritten.

To further evaluate the log file external software can be used. There are several programs which are able to read JSON files as well as interfaces to most of the programming languages.

About EffortLog

This help provides information about EffortLog.

History of EffortLog

EffortLog was started in 2015 at the IT Center of RWTH Aachen University. Its main purpose is to provide a simple program to aid developers in tracking their programming efforts. The gained data can than be evaluated for researching purposes on programming effort analysis. Please see www.hpc.rwth-aachen.de/research/tco for more information on the research done with EffortLog.

Change log:

Change log

- 1.0.0 (May 2021):
  - Added feature: Support for Qt 6
- 0.9.6 (March 2021):
  - Chore: Updated to Qt 5.15
  - Chore: Removed some deprecated code
  - Bug fix: Add ctrl key to hotkeys to prevent interference with user input
  - Bug fix: Negative time in the header when logging across date borders
- 0.9.5 (January 2019):
  - Bug fix: Missing visual updates to project settings
  - Bug fix: Wrong type when importing project files
  - Bug fix: Segfault when pressing “Log current efforts”
- 0.9.4 (June 2018):
  - Added feature: Questionnaires remember characteristics of last performance measurement
  - Bug fix: Improved scaling on high-DPI displays
- 0.9.3 (April 2018):
  - Added feature: Added a tray icon and desktop notifications
  - Added feature: The view on the questionnaire is now scrollable
  - Bug fix: Improved cross-platform window displaying
- 0.9.2 (March 2018):

  - Added feature: Simplified set-up
- 0.9.1 (March 2018):
  - Added feature: Added auto-completion to most of the questionnaire forms
  - Bug fix: Consistent scheduler IDs for the diary entries
- 0.9.0 (February 2018):
  - Added feature: Major reorganization of the questionnaires towards a single-page layout
- 0.8.3 (May 2017):
  - Bug fix: Deployment reverted to Qt5.6 due to packaging issues with QT5.8
- 0.8.2 (May 2017):
  - Added feature: Support for changes to file directories for existing projects
  - Bug fix: Skip read of log file on newly created projects
  - Bug fix: Support for blanks in directory names
- 0.8.1 (January 2017):
  - Added feature: Added a viewer of the current log file during the questionnaire
- 0.8 (September 2016):
  - Added feature: Added a convenient view of the current log file sorted by dates
  - Added feature: Can now handle development sessions spanning multiple days
- 0.7 (February 2016):
  - Added feature: Encryption is disabled by default. Add ‘-config crypt‘ to your qmake flags to enable encryption
  - Bug fix: Empty time stamps in the log files
- 0.6 (January 2016):
  - Added feature: Support for full 256-bit AES encryption of all project and log files
  - Bug fix: Appending an event fails due to encryption
  - Bug fix: Wrong interval length of the first logging event
- 0.5 (December 2015):
  - Added feature: Added milestones to projects
  - Added feature: Added IDs to each logging event
  - Bug fix: Fixed US locale to unify data format of log files
  - Bug fix: Milestones have mismatching IDs between the logging event and the current milestone
- 0.4 (December 2015):
  - Added feature: Configuration unified in one dialog
  - Added feature: Added appendix to log file at program startup
  - Bug fix: General fixes
- 0.3 (November 2015):
  - Added feature: New configuration window
  - Added feature: Main window simplified

    - Bug fix: Errors with the generated JSON files
- 0.2 (October 2015):
    - Bug fix: Improved JSON handling
- 0.1 (September 2015): Initial release

Needed Privileges to Run EffortLog

EffortLog can be run as any user as long as the user owns the rights to write to the specified output directory of the log file. Note that users on Unix operating systems might need to set the binary file to be executable. The binary file is called "effort-log" on Linux operating systems and can be set executable by the following command: `chmod +x effort-log`.

Licensing of EffortLog

Copyright © 2015-2019 IT Center, RWTH Aachen University

This project is licensed under version 3 of the GNU General Public License.

Help the Development of EffortLog

For more information on the research done with EffortLog and general questions, please see www.hpc.rwth-aachen.de/research/tco or contact Sandra Wienke or Julian Miller.

For information regarding the development of EffortLog, please see the readme file on Github or contact Julian Miller.