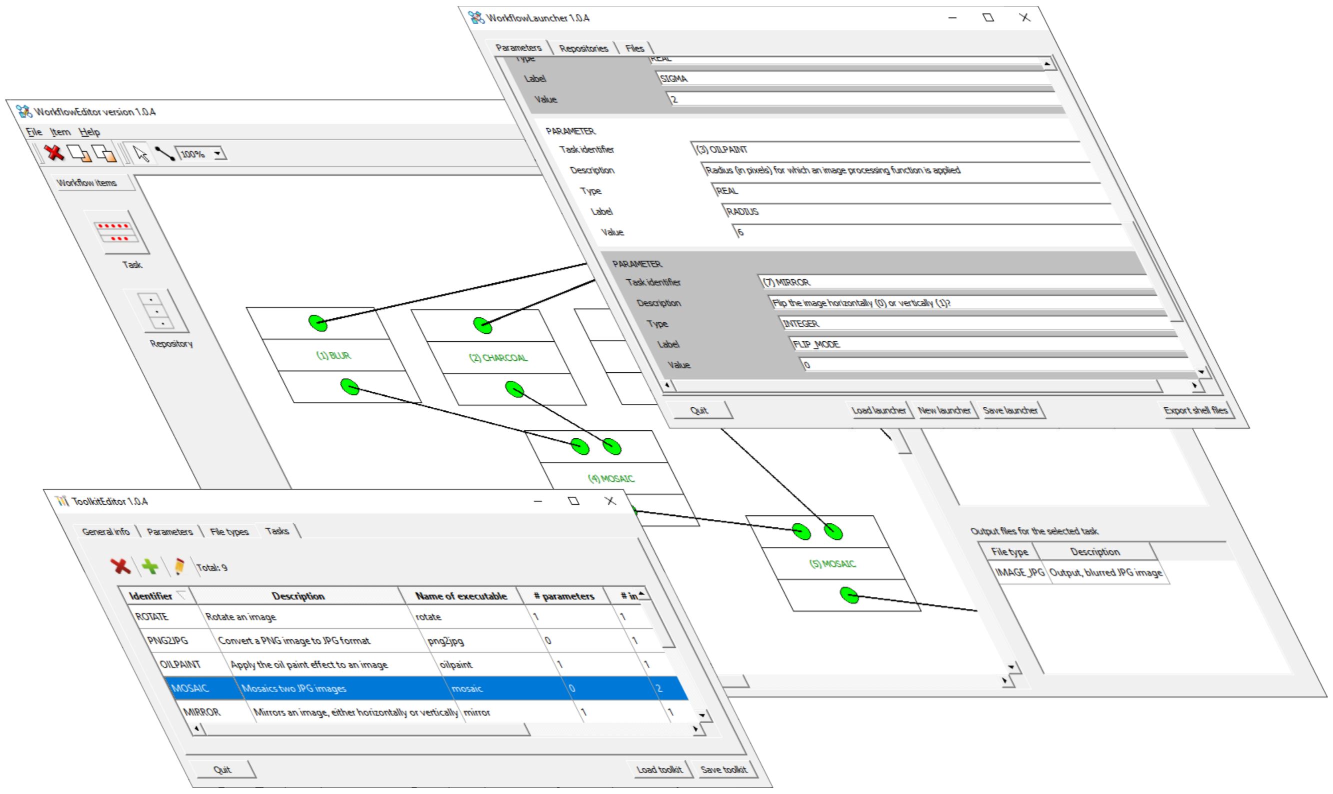
WorkflowMaker – About the samples source code

For WorkflowMaker version **1.0.5** and later
Windows 10 & 11
64-bit only



Label	SIGMA
Value	2

PARAMETER

Task identifier	(3) OILPAINT
Description	Radius (in pixels) for which an image processing function is applied
Type	REAL
Label	RADIUS
Value	6

PARAMETER

Task identifier	(7) MIRROR
Description	Flip the image horizontally (0) or vertically (1)?
Type	INTEGER
Label	FLIP_MODE
Value	0

Output files for the selected task

File type	Description
IMAGE_JPG	Output, blurred JPG image

Total: 9

Identifier	Description	Name of executable	# parameters	# in
ROTATE	Rotate an image	rotate	1	1
PNG2JPG	Convert a PNG image to JPG format	png2jpg	0	1
OILPAINT	Apply the oil paint effect to an image	oilpaint	1	1
MOSAIC	Mosaics two JPG images	mosaic	0	2
MIRROR	Mirrors an image, either horizontally or vertically	mirror	1	1

Table of contents

List of figures

1 About the samples source code

Figure 1 depicts the organization of the source code once the samples have been installed.

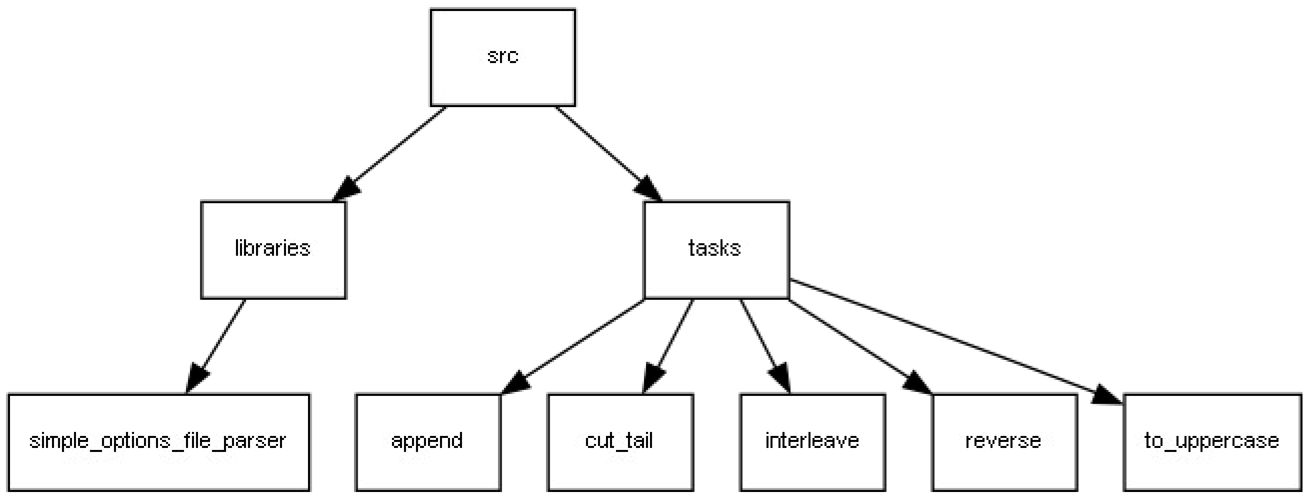


Figure 1: Folder hierarchy for the source code of the samples.

The source code is located at the "src" subfolder of the installation directory. If the standard options are accepted when running the installer, then this location is

```
C:\Program Files\cttc\WorkflowMaker_text_processing_samples
```

On Windows, it is possible to access the "src" folder by means of the start menu. There, an entry named "WorkflowMaker_text_file_processing_samples" may be found. Unfolding this entry, another one, named "WorkflowMaker_text_file_processing_samples Source code folder" may be found. Selecting this entry, the "src" folder will be opened.

As shown in Figure 1, the source code for five tasks (console applications) and one library has been included.

The library, `simple_options_file_parser,` is targeted at facilitating the parsing of options files including label + value pairs (as requested by WorkflowMaker). All tasks rely on this library to parse their corresponding options files. This library is self-contained and depends on no other components.

The five tasks are all organized in the same way, consisting always of three files:

- main.cpp.
- xxxxx_options_file_reader.cpp + xxxxx_options_file_reader.hpp.

where “xxxxx” stands for the name of any of the five tasks included. In the case of the append task, these filenames are

append_options_file_reader.cpp + append_options_file_reader.hpp

These xxxxx_options_file_reader files implement specific options file readers for each of the tasks, taking care of searching for the appropriate labels and then reading their values. Since each task reads a different set of labels, a specific options file parser is required.

For instance, the labels to parse for the append task are APPEND_INPUT_FILENAME_1, APPEND_INPUT_FILENAME_2, and APPEND_OUTPUT_FILENAME. In the case of interleave, these are INTERLEAVE_INPUT_FILENAME_1, INTERLEAVE_INPUT_FILENAME_2, INTERLEAVE_OUTPUT_FILENAME, and INTERLEAVE_COPY_QUEUE.

The main.cpp files implement the specific logic for each task. A common trait is that all these main.cpp create an object of the corresponding xxxxx_options_file_reader class to read the parameters controlling its behavior.

To build source, it is necessary to build the simple_options_file_parser first. Then, it is possible to build the tasks.