Lesson 2. Processing My First Video with λ·vue!

In this lesson, we demonstrate how to magnify a video file using default settings with **MagEngine**.

Program

```
#include <QCoreApplication>
#include <iostream>
#include <unistd.h>
/* Include the main Lambda SDK header file */
#include <MagEngineAPI.h>

using namespace std;
using namespace geko;

bool run_engine = true;

/* Callback function to handle various Lambda events */
void cbFunction(struct EventValue value) {
  switch(value.event) {
    case OPEN_FILE_SUCCESS:
      cout << "File " << getSourceName() << " opened, processing ..." << endl;
      break;
    case OPEN_FILE_FAIL:
      cout << "File open failed" << endl;
      run_engine = false;
      break;
    case END_OF_FILE:
      cout << "Process completed" << endl;
      run_engine = false;
      break;
    default:
      break;
  }
}

int main(int argc, char *argv[])
{
  QCoreApplication a(argc, argv);

  /* Display the current Lambda SDK version */
  cout << "Hello World, I am using Lambda SDK verson: "
       << MAG_ENGINE_VERSION() << endl;

  /* Initilize Lambda engine using defined Callback function as parameter
   * and return the license status (See SDK manual for states)
   */
  enum LicenseResponse license_status = initMagEngine(cbFunction);

  /* Check the license state, continue only if the license is valid */
  cout << "Lambda license status: " << license_status << endl;
  if (license_status <= 0) {
    cout << "Valid license" << endl;
    setFileOutput(RECORD_PROCESSED_ONLY);
    char video_source[] = "C:/Programming/baby.mp4";
    setSource(OPEN_FILE_SOURCE, video_source);
    /* Loop to control when to stop the Lambda engine */
    while (run_engine) {
      sleep(1);
    }
  } else {
    cout << "Invalid license" << endl;
  }

  /* Destroy Lambda Engine before exiting the program */
  destroyMagEngine();

  return a.exec();
}
```

You can download the testing video baby.mp4 here.

Line-by-line Explanation

```
#include <unistd.h>
```

Include the standard POSIX header file so we can have access to *usleep* function.

```
bool run_engine = true;
```

Boolean variabled added so we can control when to terminate the application.

```
setFileOutput(RECORD_PROCESSED_ONLY);
```

Set the engine output mode to *RECORD_PROCESSED_ONLY* (Default is *RECORD_NONE*. For complete description of modes, please refer to the λ·vue SDK API Manual). This instruct **MagEngine** to save the processed video to a file.

```
char video_source[] = "C:/Programming/baby.mp4";
setSource(OPEN_FILE_SOURCE, video_source);
```

Specify the source video file to process. Function *setSource()* instruct **MagEngine** to open the specified file **AND** immideiate start the magnification process.

```
switch(value.event) {
  case OPEN_FILE_SUCCESS:
    cout << "File " << getSourceName() << " opened, processing ..." << endl;
    break;
  case OPEN_FILE_FAIL:
    cout << "File open failed" << endl;
    run_engine = false;
    break;
  case END_OF_FILE:
    cout << "Process completed" << endl;
    run_engine = false;
    break;
  default:
    break;
}
```

Switch statement added to handle **MagEngine** events. In this lesson, we only handle events related to *setSource()*, namely:

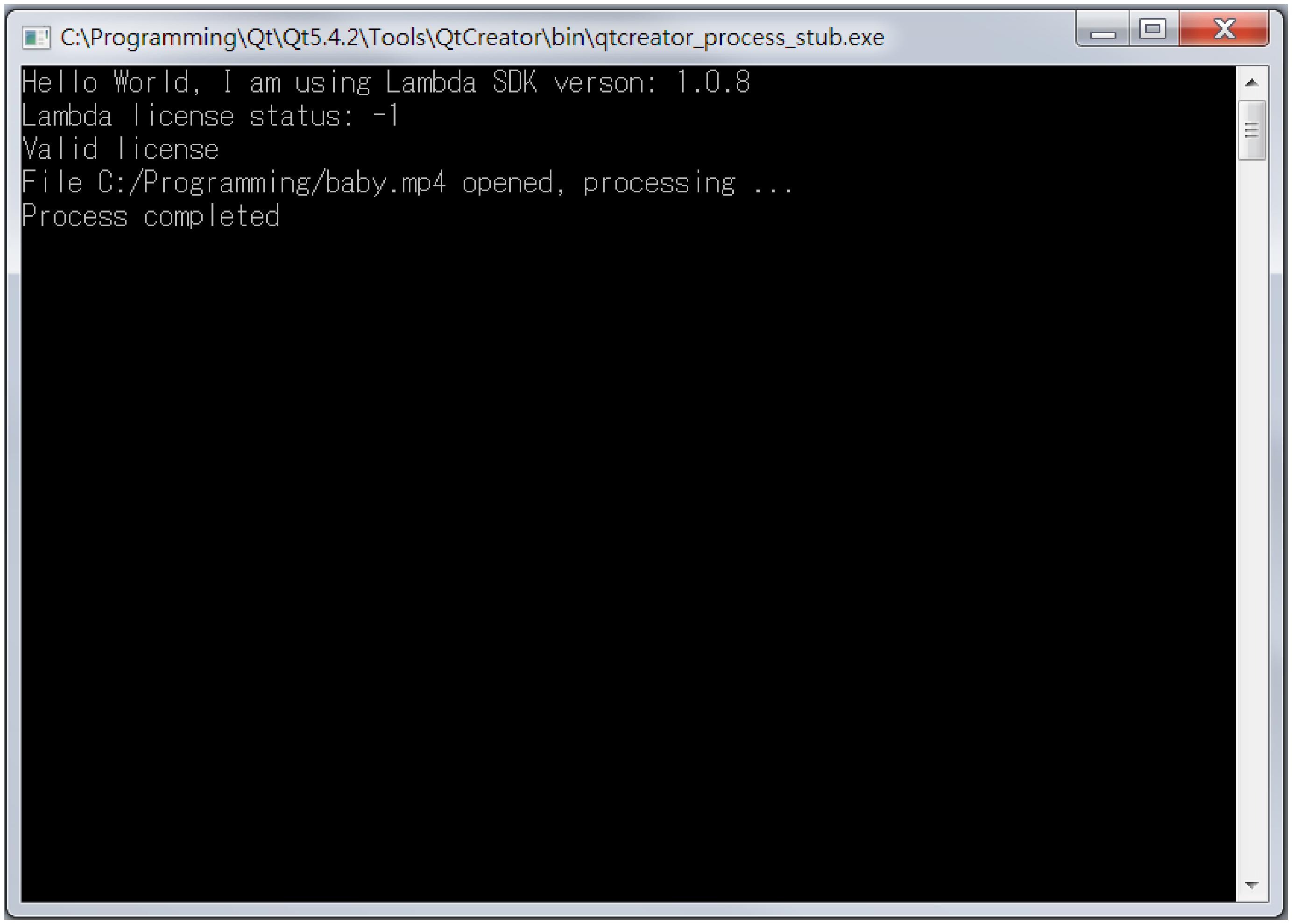
- OPEN_FILE_SUCCESS: event generated when the source file has been successfully opened by **MagEngine**
- OPEN_FILE_FAIL: event generated when **MagEngine** failed to open the specified source file. Boolean variable *run_engine* is set to false to terminate **MagEngine** after file read error.
- END_OF_FILE: event generated when end of file reached while reading source video file. Boolean variable *run_engine* is set to false to terminate **MagEngine** after video magnification is completed for the specified file.

```
while (run_engine) {
  usleep(100);
}
```

Loop to prevent program from terminating before desired break point. In this lesson, the break point is after the specified video has been processed or when **MagEngine** failed to open the specified file.

Program Output

When you run this program, you should get the following console output and the output video file under *C:/Programming* directory named *baby_TIMESTAMP_processed.avi* which you can then open with any video player to view the result.


C:\Programming\Qt\Qt5.4.2\Tools\QtCreator\bin\qtcreator_process_stub.exe

```
Hello World, I am using Lambda SDK verson: 1.0.8
Lambda license status: -1
Valid license
File C:/Programming/baby.mp4 opened, processing ...
Process completed
```

Figure 1. Lesson 1 Output