

Examples Guide

Contents

LUNA SDK Examples	3
Building Examples	3
Running Examples	3
Building Examples for Windows	4
Running on Windows:	4
Building Examples for MacOS	4

LUNA SDK Examples

Welcome to Example guide, which describes sample code for VisionLabs LUNA SDK. The presented examples are compatible with SDK version 5.0.0 and newer.

Please note that while these examples are released under MIT license, the SDK itself is not. Contact us via email (info@visionlabs.ru) for evaluation and/or licensing terms and conditions.

Aside from examples itself, there are some supplementary materials you may find useful. Look into *cmake/* folder for a CMake find script for the SDK. CMake usage is not mandatory but we advise you to do it.

Currently we support 64 bit Windows and Linux. On Windows, everything should work with Visual Studio 2015. On Linux, we tested this code with GCC 4.8.5. The other versions may work as well. Note, that the SDK is officially supported on RedHat Linux families (RHEL, CentOS, Fedora).

For more detailed information about each example, look in the corresponding directory and check out README.md

Building Examples

The build sequence for basic examples (example_extraction, example_estimation, example_descriptor_io, example_index) is presented below. The main dependency for basic examples is LUNA SDK, these examples should work out of the box. From Luna SDK root.

```
$ mkdir build && cd build
$ cmake -DCMAKE_BUILD_TYPE=Release -DFSDK_ROOT=.. ../examples
$ make
```

Optionally you can enable complementary examples, which require additionally installed 3rd party libraries:

- Liveness examples (example_liveness, example_depth) - shows interface compatibility of LUNA SDK (LivenessEngine) and OpenCV library.

Please refer to requirements sections of corresponding examples.

Running Examples

Data folder is required at <LUNA_SDK_root>/data on Windows and on Linux.

```
$ build/example_estimation/example_estimation examples/images/portrait.ppm
```

Building Examples for Windows

Building command (from FSDK_ROOT):

```
$ mkdir build install && cd build
$ cmake -G "Visual Studio 15 2017 Win64" -DCMAKE_BUILD_TYPE=Release -
  DFSDK_ROOT=.. -DCMAKE_INSTALL_PREFIX=..\install ../examples
$ cmake --build . --config Release --target install
```

After this steps You can find examples is in '`FSDK_ROOT\install\bin\`' directory.

Running on Windows:

Manually register the *necessary dynamic libraries* (.dll) in the system (if needed).Or copy all necessary dynamic libraries (*.dll) in the directory with an example.

Run the command from "FSDK_ROOT".

```
$ install\bin\example_estimation.exe examples/images/Jennifer_Aniston.jpg
```

LivenessEngineSDK.dll, *FaceEngineSDK.dll*, *tbb.dll*, *flower.dll*, **TrackEngineSDK.dll*

Running on Windows (from FSDK_ROOT):

Note: Manually register the *necessary dynamic libraries* (.dll) in the system (if needed).Or copy all necessary dynamic libraries (*.dll) in the directory with an example.

```
$ install\bin\example_estimation.exe examples/images/Jennifer_Aniston.jpg
```

LivenessEngineSDK.dll, *FaceEngineSDK.dll*, *tbb.dll*, *flower.dll*, **TrackEngineSDK.dll*

Building Examples for MacOS

Building command (from FSDK_ROOT):

```
$ mkdir build && cd build
$ cmake -DCMAKE_BUILD_TYPE=Release -DFSDK_ROOT=.. ../examples
$ make
```

If you have some problemes with execution you will possibly need to re-sign all binary files. For example you can do it by such a way:

```
codesign -v --force --sign - --entitlements test.xcent lib/clang/x64/*
codesign -v --force --sign - --entitlements test.xcent bin/clang/x64/*
```

Possible content of test.xcent.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/
  DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

  <dict>

    <key>com.apple.security.get-task-allow</key>

    <true/>

  </dict>

</plist>
```

Also you possibly should to allow execution for binary file on your device: System Preference - Security & Privacy - General - Allow apps downloaded from anywhere.