

# Quick Start Guide

## Contents

<b>Quick Start Guide</b>	<b>3</b>
<b>SDK package structure</b>	<b>4</b>
<b>Getting started</b>	<b>5</b>
Bundled face descriptor model versions . . . . .	6
Bundled reidentification descriptor model version . . . . .	7
<b>Where is the code</b>	<b>8</b>

## Quick Start Guide

This short guide will help you to get started with the SDK. Luna SDK contains two primary libraries:

- FaceEngineSDK;
- LivenessEngineSDK.

FaceEngineSDK is the core library. It allows to implement the following functionality:

- Faces detection;
- Faces normalization (warping);
- Landmarks location on faces;
- Face descriptor extraction and matching;
- Descriptor search index (fast Hierarchical Navigable World graph or HNSW shortly);
- Estimators :
  - Attributes estimator (age, sex, glasses presence);
  - Quality estimator;
  - Black/White estimator;
  - Eyes estimator (opened/closed eyes);
  - Head pose estimator.
  - Gaze estimator.
  - Smile estimator.
  - Emotions estimator.
  - Child estimator.
  - Overlap estimator.
  - Credibility check estimator.
  - Facial hair estimator.

*FaceEngineSDK* library is available in two versions: Frontend Edition (FE) and Complete Edition (CE):

- Complete Edition supports all functionality;
- Frontend Edition is a light version with excluded descriptor functionality (does not support descriptor extraction and matching).

LivenessEngineSDK\_ library is designed to distinguish a real user's face from a fake one (e.g., printed on paper or displayed on the screen). LivenessEngine uses FaceEngineSDK library modules and performs detection, warping, and various estimations.

## SDK package structure

Folder	Contents
/include	C++ header files
/lib/<compiler>/<bitness>	Libraries (.lib on Windows, .so on Linux)
/bin/<compiler>/<bitness>	Binaries and tools (also .dll files on Windows)
/doc	Documentation
/data	Algorithm model data required to use the SDK
/examples	Sample code

**Note 1:** <compiler> is “gcc4” on Linux and “vs2015” on Windows.

**Note 2:** GCC versions 5 and later break backward compatibility with GCC4, so do not mix them.

**Note 3:** <bitness> is “x64” for 64-bit OS and “x86” for 32-bit OS.

## Getting started

It is recommended to familiarize with the common FaceEngine SDK concepts and terminology first. For that, please refer to the handbook in **/doc/FaceEngine\_Handbook.pdf**. Handbook contents is as follows:

- Chapters 1-3 cover common concepts and FaceEngine SDK modules;
- Chapters 4-7 describe each module in depth;
- Chapter 8 tells more about licensing;
- Chapter 9 is dedicated to hardware and software requirements. Please make sure your system meets the requirements listed in chapter 9 before proceeding.

In appendixes one may find performance evaluation results and answers to some frequently asked questions.

Note, that the purpose of the handbook is to describe common concepts and give an idea what LUNA SDK is capable of. For detailed descriptions of particular functions, refer to the reference manual instead. FaceEngine reference manual (doxygen) is available in the **doc/fsdk/html/** directory.

FaceEngine SDK configuration parameters are described in **doc/ConfigurationGuide.pdf**.

Liveness SDK reference manual (doxygen) is available in the **doc/lsdk/html/** directory.

For additional information on Liveness SDK refer to the liveness handbook in **doc/LivenessEngine\_Handbook.pdf**.

Document **doc/Featuremap.htm** contains information about matching between SDK facilities and data files which these facilities needs.

## **Bundled face descriptor model versions**

The SDK implements several face descriptor models.

Multiple models are maintained for backward compatibility reasons, as cross-model matching is not supported.

The recommended version for new users is 54, which works identically on all supported platforms. You may find additional details on descriptors in Chapter 7 and Appendix A of the SDK Handbook.

### **Bundled reidentification descriptor model version**

The SDK implements one human descriptor model. The start version for reidentification (human) descriptors is 101. To create descriptors, batchs, matcher and extractor of human type you must pass version of human descriptor.

## Where is the code

The example code for server and PC versions is located in the **/examples** folder. Please refer to doc/ExamplesGuide.pdf, for additional information.