



VisionLabs LUNA KIOSK

v.2.6.0

None

1. Glossary	4
2. Introduction	5
3. System requirements	6
3.1 Minimum system requirements for x64 architecture	6
3.2 Minimum system requirements for ARM architecture	6
3.3 Supported cameras	7
3.3.1 Connection requirements	7
3.4 Orbbec and Intel RealSense SDK compatibility	7
3.4.1 Orbbec SDK v.1.10.27	7
3.4.2 Orbbec SDK v.2.5.5	8
3.4.3 Intel RealSense SDK v.2.56.5	9
4. Licensing	11
4.1 Switching the license type	11
5. Installation	12
5.1 Installation on Windows	12
5.1.1 Step 1: License activation	12
5.1.2 Step 2: Preparing for LUNA KIOSK installation	13
5.1.3 Step 3: System configuration	13
5.1.4 Step 4: System management	14
5.1.5 Delayed service start	14
5.1.6 Troubleshoot metadata sampling issues for RealSense camera sensors	14
5.2 Installation on Ubuntu 24.04 x64, Debian 10 x64, and Armbian 23 operating systems	14
5.2.1 Prerequisites	14
5.2.2 Step 1: Pre-action and license activation	15
5.2.3 Step 2: LUNA KIOSK installation	17
5.2.4 Step 3: LUNA KIOSK setup	18
5.2.5 Step 4: LUNA KIOSK management in Linux	18

5.3	Offline license activation	18
5.3.1	Prerequisites	18
5.3.2	Step 1: Generate device fingerprint	18
5.3.3	Step 2: Activate license	19
5.3.4	Step 3: Deploy license	20
6.	Deletion	21
6.1	Uninstall on Windows	21
6.2	Uninstall on Ubuntu 24.04 x64, Debian 10 x64 and Armbian 23	21

1. Glossary

Term	Definition
Bestshot	The frame of the video stream in which the face is captured in the best angle for further use in a face recognition system
Liveness	Software method that enables you to confirm whether a person in one or more images is "real" or a fraudster using a fake ID (printed face photo, video, paper or 3D mask)
Detection	Actions to find areas of the image containing faces
Spoofing attack	Substitution of a real person for a fake image (for example, a photograph) to deceive the system

2. Introduction

This document describes:

- the process of installing as well as uninstalling the VisionLabs LUNA KIOSK application,
- hardware and software requirements of the equipment.

VisionLabs LUNA KIOSK (hereinafter referred to as System) is a set of libraries that provide the possibility of realizing real-time operation to perform face detection in a frame, check the vitality of a person and transfer data to an external system.

The System is designed for:

- receiving and processing a color video stream from a video recording device,
- checking the image quality,
- selecting the bestshot,
- face detection by machine calculation method on two images,
- checking the presented image by Liveness-algorithms,
- protection against image spoofing by depth map analysis,
- subsequent transfer of the bestshot to device integration systems.

3. System requirements

To successfully install and run the system, your hardware must meet the minimum requirements specified below.

3.1 Minimum system requirements for x64 architecture

Required Resource	Recommended
Processor	Intel(R) Core(TM) i3-10110U
RAM	4GB or more
Hard disk drive	HDD or SSD at least 1,4 B
Operating system	<ul style="list-style-type: none">• Windows 10 (64 bit)• Ubuntu 24.04 x64• Debian 10 x64
Instruction Support	Advanced Vector Extensions 2 (AVX2)

To run the application on Windows, install the [Visual C++ Redistributable package](#).

3.2 Minimum system requirements for ARM architecture

Required Resource	Recommended
Processor	Rockchip RK3588S
RAM	4GB or more
Hard disk drive	HDD or SSD at least 128GB
Operating system	Armbian 23 (aarch64)

3.3 Supported cameras

The system is optimized and tested for reliable operation with the following 3D and IR cameras:

- **Intel® RealSense™ Depth cameras:**

- D415
- D435
- D435j
- D455

- **VLS LUNA CAMERA:**

- VLS LUNA CAMERA 3D / VLS LUNA CAMERA 3D Embedded
- VLS LUNA CAMERA 2D

- **Orbbec cameras:**

- Gemini E
- Gemini 335

Important: Before using Orbbec cameras with LUNA KIOSK, ensure that the official Orbbec driver is installed on the target system. Without the driver, the camera may not be recognized by the operating system or the SDK. Refer to the [Orbbec Developer Portal](#) for the latest driver packages and installation instructions.

Note: For information about VLS LUNA CAMERA 3D / VLS LUNA CAMERA 3D Embedded, please contact your VisionLabs representative.

3.3.1 Connection requirements

For stable performance, connect all supported cameras using USB 3.0.

3.4 Orbbec and Intel RealSense SDK compatibility

This section provides compatibility information for Orbbec and Intel RealSense SDKs with their respective hardware and operating systems.

3.4.1 Orbbec SDK v.1.10.27

- **Hardware compatibility:**

- Gemini E
- Gemini 335

- **Firmware requirements:**

- Minimum firmware version: 1.6.00

- **Operating system support:**

- **Windows:**

- Windows 10 or later
- Architecture: x86 and x64

- **Linux:**

- Architecture: x64 only
- Recommended distributions:
 - Ubuntu 20.04 LTS
 - Ubuntu 22.04 LTS
 - Ubuntu 24.04 LTS

3.4.2 Orbbec SDK v.2.5.5

- **Hardware compatibility:**

- Gemini E
- Gemini 335

- **Firmware requirements:**

- Minimum firmware version: 1.6.00

- **Operating system support:**

- **Windows:**

- Windows 10 or later
- Architecture: x64 only

- **Linux:**

- Architecture: x64 only
- Recommended distributions:
 - Ubuntu 20.04 LTS
 - Ubuntu 22.04 LTS
 - Ubuntu 24.04 LTS

3.4.3 Intel RealSense SDK v.2.56.5

- **Hardware compatibility:**

- Cameras:

- Intel RealSense D415
 - Intel RealSense D435i
 - Intel RealSense D435

- **Firmware requirements:**

- Minimum firmware version: 5.17.0.10

- **Operating system support:**

- **Windows:**

- Supported versions:

- Windows 11
 - Windows 10 (Build 15063 or later)

- Recommended configurations:

- Windows 10 RS5 (Redstone 5)
 - Windows 11 KB5030219 (OS Build 22621.2283 or later)
 - Build 17763+ recommended for all installations

- **Linux:**

- Supported distributions:

- Ubuntu 20.04 LTS
 - Ubuntu 22.04 LTS
 - Ubuntu 24.04 LTS
 - Non-LTS distributions are not supported

- Kernel versions:

- 6.[2, 5, 8]
 - 5.[0, 3, 4, 8, 13, 15, 19]

Hardware restrictions

The following Intel chipsets are not supported:

Intel Chipset Series	PCH Chip-ID	Compatible CPUs
300 Series/C240	9DED, A36D	8th/9th Generation Core Mobile/Desktop
600 Series	51ED	12th Generation Core Mobile

4. Licensing

To obtain a license, please contact your VisionLabs representative. The response will include the necessary license activation parameters.

Note: You can use the same license activation data across all supported platforms, including Windows, Ubuntu, Debian, and Armbian. For the license activation process, see the [Installation](#) section.

4.1 Switching the license type

You can configure the system to use either FIT or Zeus license by modifying the `licenseModel` parameter.

To do this, locate the `licenseModel` parameter in the `data/license.conf` file and assign one of the following values:

- `1` to use the FIT license.
- `2` to use the Zeus license.

Once configured, save the file and restart the service to initialize the new license type.

5. Installation

LUNA KIOSK is distributed in two configurations, differing in how configuration settings are stored:

- **On Ubuntu 24.04 x64, Debian 10 x64, and Armbian 23:** Configuration files *server.conf* and *rsengine.conf* located in the */client* directory are used.
- *server.conf* - Contains server operation and logging settings.
- *rsengine.conf* - Defines thresholds for Liveness estimations.
- **On Windows:** Settings are stored in the Windows Registry by default after installation.

Note: The Windows distribution does not include configuration files in the */client* folder, as registry-based configuration is recommended. If you require configuration files, please request them from VisionLabs.

5.1 Installation on Windows

RSE Server is installed as a Windows service.

5.1.1 Step 1: License activation

Before installation, configure the license by editing the *data/license.conf* file with values provided in your license letter from VisionLabs.

The table below outlines the parameters required for license activation:

Parameter	Description	Example
<code>Server</code>	The URL of the activation server used to validate and activate the license.	https://
<code>EID</code>	A unique access authorization identifier.	00000000-0000-0000-0000-0000-0000-000000000000
<code>ProductID</code>	The specific product identifier.	00000000-0000-0000-0000-0000-0000-000000000000

Parameter	Description	Example
<code>LICENSE_FILENAME</code>	A license file name. Non-public parameter, do not change.	<i>license.dat</i>
<code>LICENSE_CONTAINERMODE</code>	License execution mode: <code>0</code> - run in container <code>1</code> - local (recommended)	<code>1</code>
<code>licenseModel</code>	Defines the license to be used: <code>1</code> - The system uses the FIT license. <code>2</code> - The system uses the Zeus license.	<code>2</code>

5.1.2 Step 2: Preparing for LUNA KIOSK installation

Run `InstallService.bat` as Administrator from the root of the distribution package.

During installation:

The service is registered in the Windows Registry.

A log directory is created at `C:\RSE\logs` by default.

You can adjust logging levels later in the Windows Registry.

5.1.3 Step 3: System configuration

All system parameters are written to the Windows Registry. The parameters are set to default values and can be customized by an administrator. For a full list of parameters, see [Appendix 1](#).

5.1.4 Step 4: System management

After the installation is complete, you can manage RSE Sever using the *services.msc* service manager.

Important: Do not change the location of RSE Server and *.conf files after installation. Otherwise, the system functionality will be impaired.

5.1.5 Delayed service start

When installing LUNA KIOSK, you can control whether the service will start immediately after installation.

To do this, use the `StartService` parameter of the *installService.bat* installation script. The parameter has the following values:

Value	Description
<code>true</code>	Default. The service will be automatically started after installation.
<code>false</code>	The service will be installed, but its startup will be skipped.

5.1.6 Troubleshoot metadata sampling issues for RealSense camera sensors

Windows requires a special registry entry for each unique video device to provide metadata. Metadata, or metadata attributes, refer to additional information provided by the `librealsense` library and are necessary for proper interaction with RealSense camera sensors.

To activate the functionality related to metadata attributes, the *realsense_metadata_win10.ps1* file is included with LUNA KIOSK. A script in the file creates and modifies entries in the Windows registry to ensure that the RealSense camera sensors work correctly. For more information on activating metadata, see [librealsense library documentation](#).

5.2 Installation on Ubuntu 24.04 x64, Debian 10 x64, and Armbian 23 operating systems

5.2.1 Prerequisites

Use Ansible to automate installation on Linux-based systems:

1. Install Ansible:

```
apt-get install ansible
```

2. Verify the installation:

```
ansible --version
```

If the installation is successful, the console will display the Ansible version and other information:

```
ansible [core 2.12.4]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/vivek/.ansible/plugins/modules',
  '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/vivek/.ansible/collections:/usr/share/
ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Mar 15 2022, 12:22:08) [GCC 9.4.0]
  jinja version = 2.10.1
  libyaml = True
```

5.2.2 Step 1: Pre-action and license activation

Before running the installation:

1. Add addresses (`ip` or `hostname`) of target devices to the `ansible/hosts` file to install the distribution. Each line contains one address. For example:

```
[rse]
12.16.58.33
```

2. Adjust the shared variables and license data in the `group_vars/all.yml` file:

Note: `LICENSE_SERVER`, `LICENSE_EID`, `LICENSE_PRODUCTID` are responsible for licensing and provided by VisionLabs.

Variable	Description	Possible values
<code>RSE_HOME</code>	Installation directory. Non-public value, do not change.	<code>/var/lib/kiosk</code>

Variable	Description	Possible values
<code>RSE_VERSION</code>	System version. The value must match the archive name.	<i>ub1804_x64_v1.0.4_rc2</i>
<code>RSE_ZIP_LOCATION</code>	Absolute or relative path to the RSE Server distribution within the distribution package.	<i>../distr/rse-server_v.2.5.0.zip</i>
<code>LICENSE_SERVER</code>	License server address. Request data from VisionLabs.	<i>https://.com</i>
<code>LICENSE_EID</code>	Access license identifier. Request data from VisionLabs.	<i>00000000-0000-0000-0000-0000-000000000000</i>
<code>LICENSE_PRODUCTID</code>	Product identifier. Request data from VisionLabs.	<i>00000000-0000-0000-0000-0000-0000-000000000000</i>

Variable	Description	Possible values
<code>LICENSE_FILENAME</code>	License file name. Non-public parameter, do not change.	<i>license.dat</i>
<code>LICENSE_CONTAINERMODE</code>	License execution mode.	<code>0</code> - run in container <code>1</code> - local (recommended)
<code>licenseModel</code>	License type. Request data from VisionLabs.	<code>1</code> - The system uses the FIT license. <code>2</code> - The system uses the Zeus license.

5.2.3 Step 2: LUNA KIOSK installation

The installation location of the system is specified in the `RSE_HOME` variable.

Navigate to the *ansible* folder and run:

- **For remote hosts:**

If the installation is performed on several devices at once (more than one IP address is specified in the host), run the command:

```
ansible-playbook -I hosts install_rse.yml
```

During installation, Ansible attempts to connect as the system user (root by default) via the appropriate SSH key pair. If a password is used instead of keys, add the `--ask-pass` flag to the install command.

- **For local installation:**

If the installation is performed locally on one device, run the command:

```
ansible-playbook -i hosts--connection=local--inventory 127.0.0.1, install_rse.yml
```

5.2.4 Step 3: LUNA KIOSK setup

When starting the RSE Server, the system uses the settings from the `server.conf` and `rsengine.conf` configuration files. For setting descriptions, see [Appendix 1](#).

If you need to change the client configuration settings, make changes in the `server.conf` and `rsengine.conf` files and restart the RSE Server.

If the installation is successful, the logs will be written to log files in the default `./logs` directory after LUNA KIOSK is launched. To change the location, update the `log-path` parameter in the `server.conf` file.

5.2.5 Step 4: LUNA KIOSK management in Linux

After the installation is complete, manage RSE Server using the `systemctl` command line utility.

To start RSE Server, run the following command:

```
systemctl start kiosk
```

To stop the RSE Server, run the following command:

```
systemctl stop kiosk
```

5.3 Offline license activation

Use offline activation when the target device has no internet access. In this scenario:

- The device fingerprint is generated on the offline device.
- The license is activated on an internet-connected device using that fingerprint.

5.3.1 Prerequisites

Before you start, obtain the license server address and EID from VisionLabs.

5.3.2 Step 1: Generate device fingerprint

On the offline device where the license will be installed:

1. Open the `license.conf` file located in the `data` directory. 2. Specify `EID` and save the file. 3. Run the `FingerprintViewer` utility to generate the fingerprint:

- On Linux/Armbian:

1² Navigate to the *extras* folder.

2³ Grant access rights to the *FingerprintViewer* utility:

```
```bash
 chmod +x FingerprintViewer
```
```

3⁴ Run the utility by specifying the path to the **license.conf** file:

```
```bash
./FingerprintViewer ../data/license.conf
```
```

- **On Windows:**

Run *FingerprintViewer.exe* from the root folder of the distribution package.

4⁵ Copy the displayed fingerprint from the console for use in the next step.

5.3.3 Step 2: Activate license

On an internet-connected device:

Go to the license activation portal (URL obtained from VisionLabs).

Log in using your EID.

Paste the copied device fingerprint to activate the license.

Download the generated license certificate (*licenseFile.v2c*).

Move *licenseFile.v2c* to the *data* directory.

Rename the file using one of the following methods:

Change the `Filename` parameter in the *license.conf* file according to the example:

```
<param name="Filename" type="Value::String" text="licenseFile.v2c"/>
```

Rename *licenseFile.v2c* to *license.dat* and ensure the `Filename` parameter in the *license.conf* file matches the actual name:

```
<param name="Filename" type="Value::String" text="license.dat"/>
```

5.3.4 Step 3: Deploy license

Copy the obtained *license.dat* license key to the *data* directory.

6. Deletion

6.1 Uninstall on Windows

To uninstall RSE Server and logs on Windows OS, run the `uninstallService.bat` batch file, which is located in the root folder of the distribution.

The `uninstallService.bat` batch file:

- removes all settings from the registry;
- stops and removes the RSE Server service.

The logs folder is not automatically deleted, it must be deleted manually.

6.2 Uninstall on Ubuntu 24.04 x64, Debian 10 x64 and Armbian 23

Uninstalling RSE Server and logs on Ubuntu 24.04 x64, Debian 10 x64 and Armbian 23 is done manually by the administrator, there is no executable batch file supplied. This requires:

Delete data from the `RSE_HOME` directory;

Delete the `systemd service` file from the `/etc/systemd/system/system/kiosk.service` directory.