Sentry Firefly

Installation Guide

Version 2026R1

04.02.2026

Copyright, trademarks, and disclaimer

Trademarks

Sentry is a trademark or registered trademarks of IntelexVision. All other trademarks mentioned in this guide are the property of their respective holders. This product may make use of third-party software for which specific terms and conditions may apply.

All names of people and organizations used in the examples in this text are fictitious. Any resemblance to any actual organization or person, living or dead, is purely coincidental and unintended.

Disclaimer

This document is intended for general Information purposes only and due care has been taken in its preparation. Any risk arising from the use of this information rests with the recipient, and nothing herein should be construed as constituting any kind of warranty.

IntelexVision may make changes without prior notice.

This product may make use of third-party software for which specific terms and conditions may apply.

Introduction

Scope

This document provides step-by-step instructions for installing **Sentry Firefly** and the ***Deep Learning*** models required by the system including their optimization. It includes all the necessary details to guide users through the installation process.

The scope of this guide is limited to the installation phase and does not include advanced configurations or integration with external systems.

Audience

The following roles are the intended audience for this guide:

- **System Administrators:** Responsible for managing the infrastructure and ensuring the software is installed correctly within the organization's IT environment.
- **IT Support Engineers:** Tasked with resolving technical issues during installation and initial setup.
- **Solution Architects:** Overseeing the installation to ensure alignment with the organization's architecture and project requirements.
- **Technical Implementation Specialists:** Handling the detailed implementation of the software for specific use cases.
- **Product Specialists:** Verifying the successful setup and ensuring that the installed component meets the business needs.

Pre-requisites

Technical Knowledge

- **Operating System Proficiency.** Familiarity with Windows-based systems, including navigating the interface, using command-line tools (e.g., PowerShell), and managing system settings.
- **Networking Basics.** Understanding of IP addressing, ports, firewalls, and network protocols commonly used in enterprise environments.
- **Web Interfaces** and b**rowser-based Configuration Tools**. Understanding how to navigate, manage, and customize settings within web-based platforms.
- **Video Surveillance Systems knowledge**. Understanding how security cameras, recording devices, and monitoring software work together to provide surveillance and protection.
- **Troubleshooting Readiness.** Knowledge of basic troubleshooting steps, such as interpreting log files or checking service statuses.

System Access

- **Administrator Privileges.** Access to a user account with administrator rights on the target Windows machine.
- **Network Permissions.** Make sure you have the required permissions to install software and adjust network settings, such as setting up firewall exceptions.

Preparatory Tasks

- **Review Documentation.** Familiarize yourself with this guide and any other documentation.
- **Backup Policies.** Make sure there's a backup plan ready for any existing software or data that could be impacted during installation.
- **Dependencies Check.** Verify that all required software components or services (e.g., runtime environments, frameworks) are pre-installed or accessible.

Deployment Note

Windows

Sentry Firefly on Windows contains ONE installer file.

It is highly recommended to also install the Deep Learning models for a full experience.

These models will be automatically installed during the installation of **Sentry Firefly** by calling ***Sentry Models*** installer, this installer can be called at any time as standalone installation and select a folder where the user would like to place the ***Deep Learning*** models.

The user will only pick which models will be suitable for the available hardware (Openvino and/or NVIDIA).

The flow for installation can be depicted as:

Prepare Installation

To begin the installation process, you will need to download *Sentry Firefly Installers* under latest Release available.

The Sentry software can be downloaded from the official Download Portal. Navigate to the portal, locate the product for which you need the installation files, and select the latest available release. After downloading the software, ensure it is saved and readily accessible on the appropriate servers for installation.

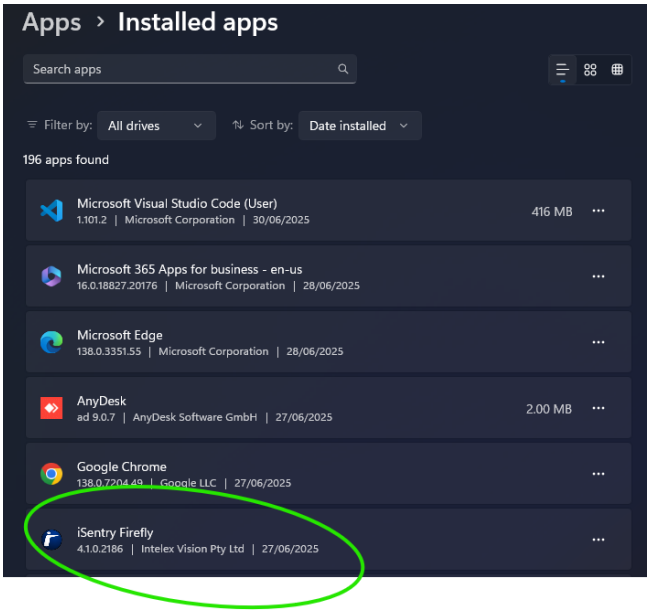
Installation steps

Windows Installation

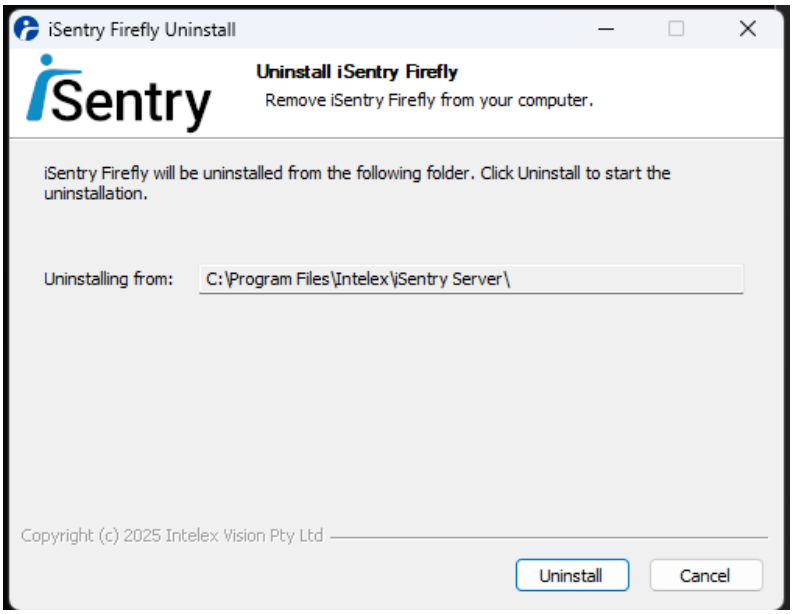
Uninstall previous version

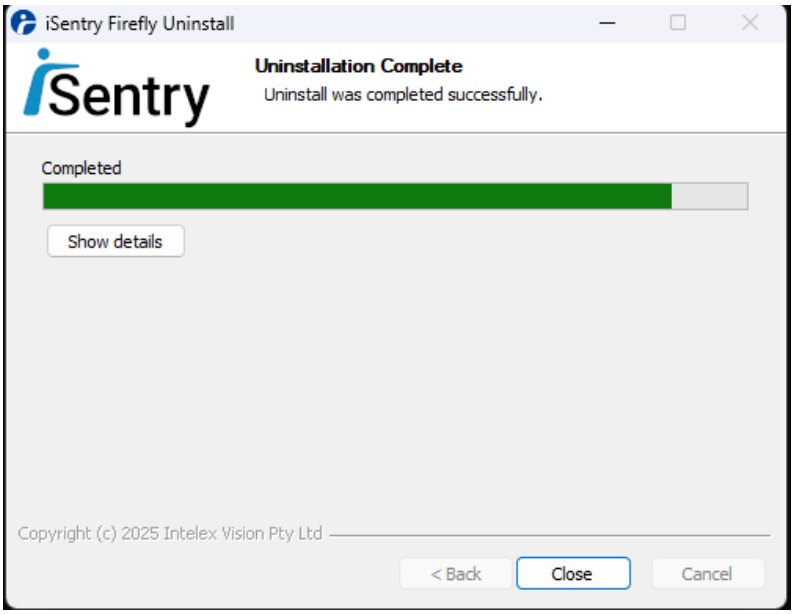
> **INFORMATION**
>
> Please Skip this step if installing **Sentry Firefly** on a fresh machine.

Go to Add/Remove files and select **Sentry Firefly**.



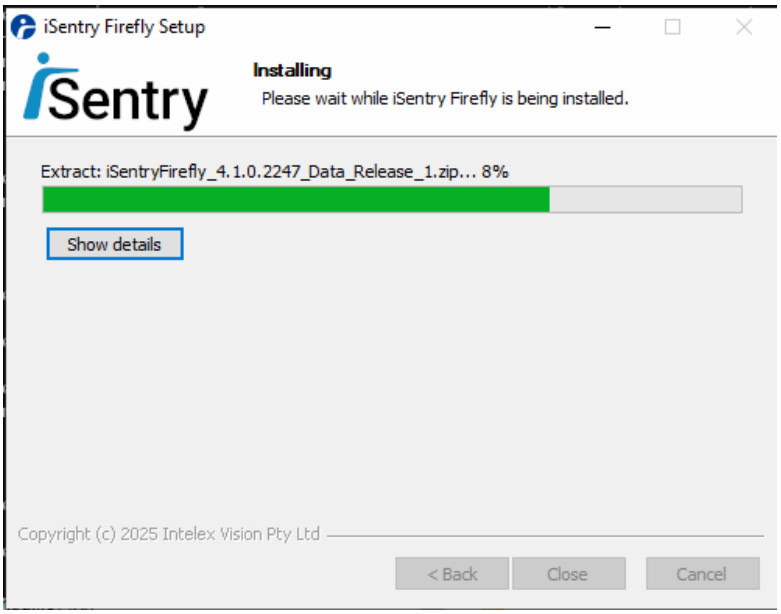
Click uninstall and wait until the process is complete,





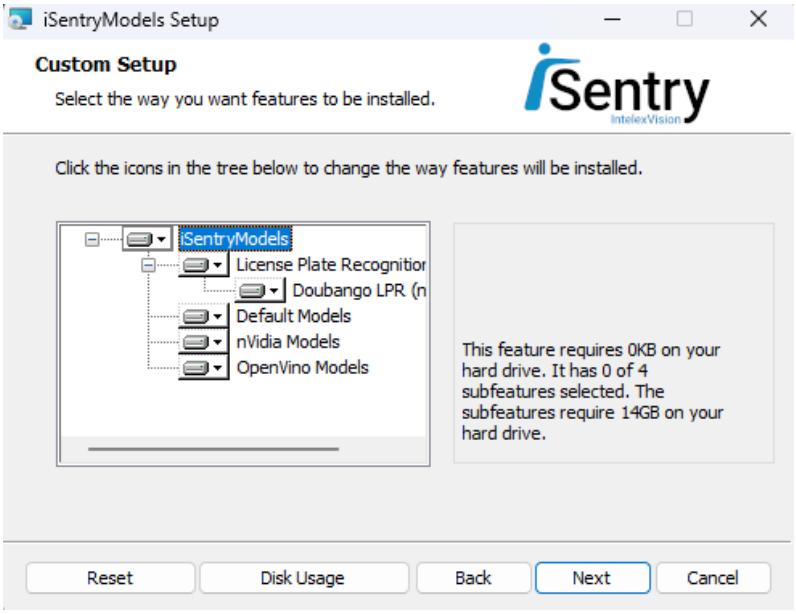
Install the standard Sentry Firefly build

Double-Click the installation exe file (e.g.: ***iSentry_Firefly_v4_1_0_2186_x64)*** and follow the setup wizard.



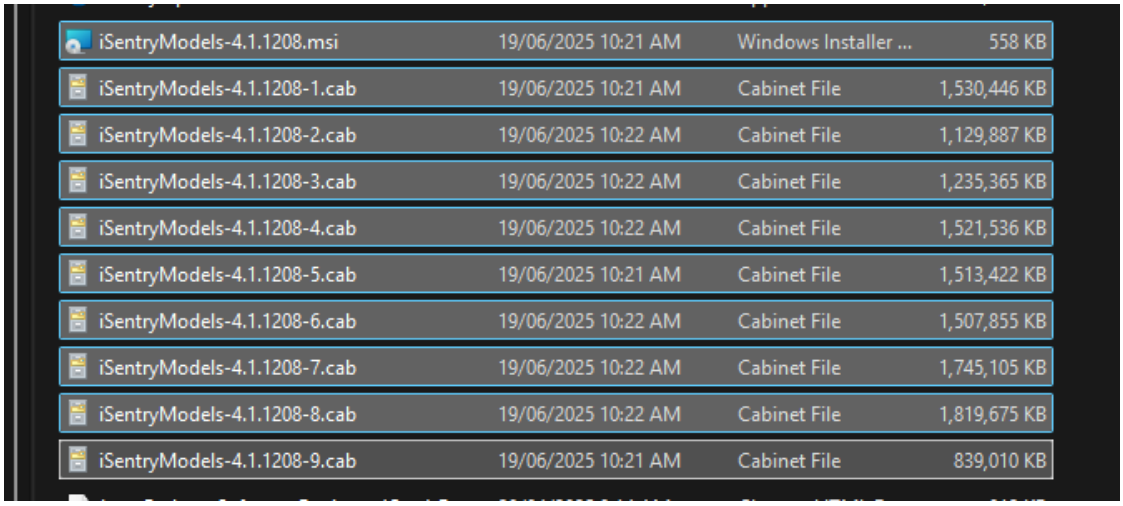
Sentry Models setup will be launched automatically.

You must select the type of installation determined by your hardware (Openvino or NVIDIA) and whether you will be using License Plate Recognition (LPR) module.



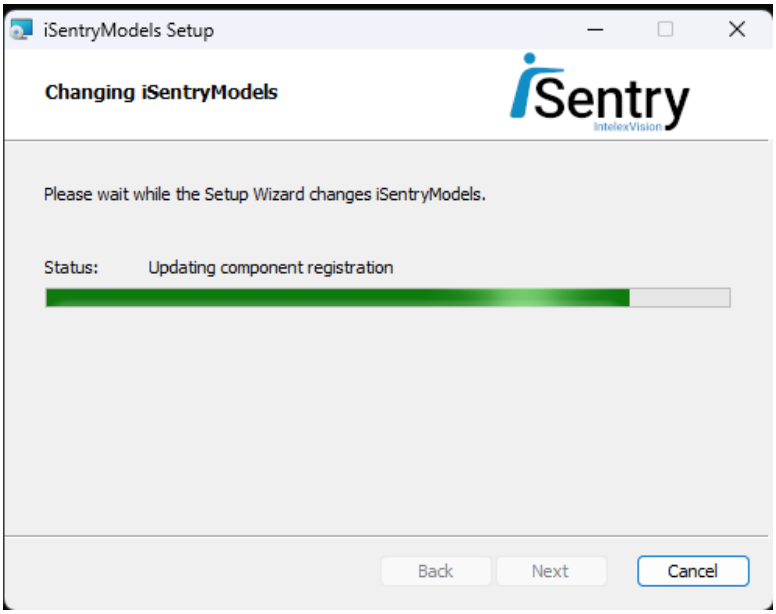
INFORMATION

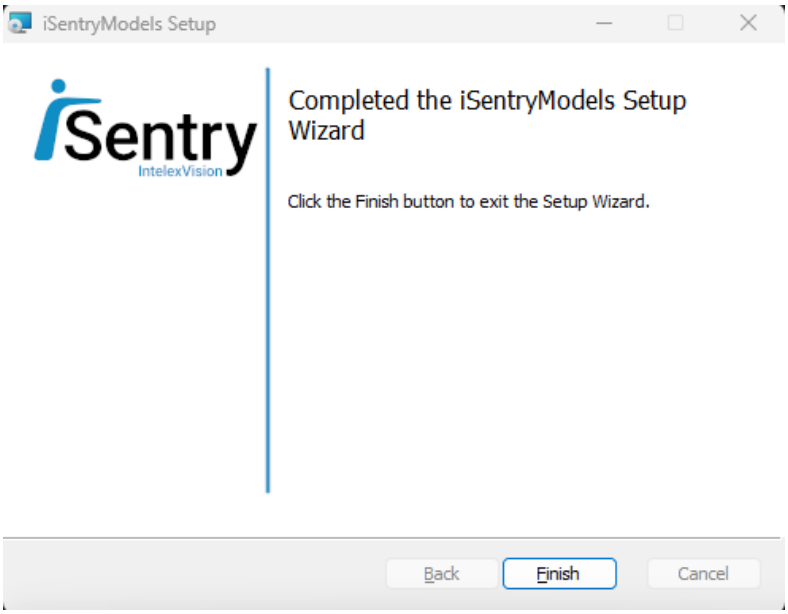
Keep in mind additional files are required to be present in the same directory as the installer, these files are the ones provided by the ***Sentry Models*** installer (the so-called .cab files, or cabinets).



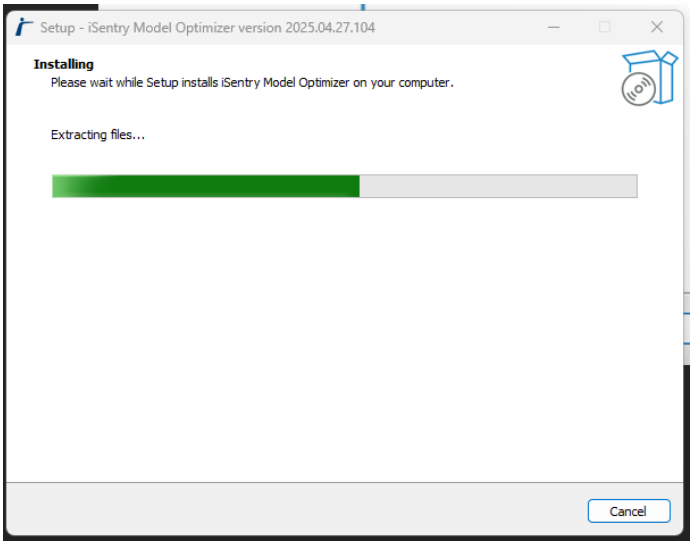
These files are required to progress with the installation of **Sentry Firefly**.

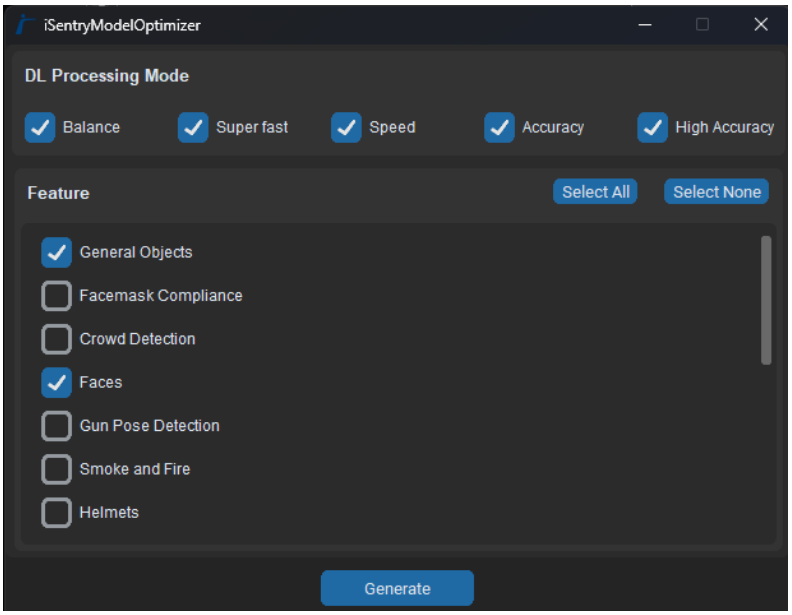
Wait for the installer to finish.





After few minutes you will be presented with the **Sentry Models Optimization**.





IMPORTANT NOTE

This tool allows you to optimize your models and accelerate their execution even further. Please keep in mind that building the engine for your models for your nVidia GPU is time consuming.

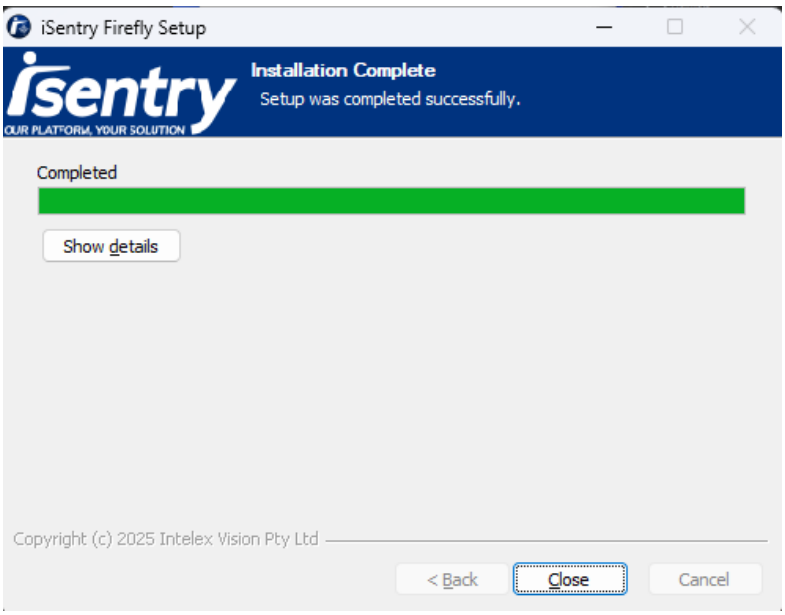
Allow more than 2 hours for all of them to be generated.

INFORMATION

You can skip this or come back later to the **Model Optimizer** tool. To close it and finish the installation just close the window pressing X icon and **Sentry Firefly** will start running.

IMPORTANT NOTE

Depending on the **Sentry Firefly** version you are upgrading from your you may or may not need to optimize the models again. The tool will check it for you, if you cannot wait, again, you can close it and come back later to generate the optimize version of your models**.**



See the icon tray. Eventually the icon will turn green.

Post Installation

GPU driver check

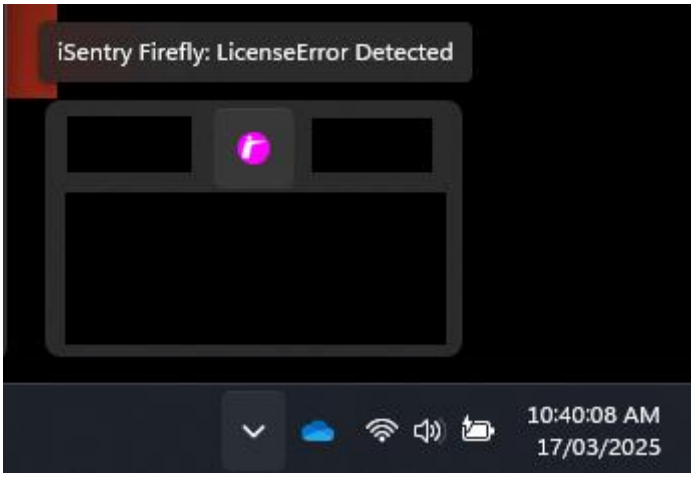
Make sure the latest nVidia driver has been installed (currently supporting CUDA 12.8 or 13.0). If the system is in Intel Graphic Card, please make sure the latest drive is installed.

Firewall Check

> **IMPORTANT NOTE**
>
> Make sure the inbound port ***8123*** and ***8124*** are allowed or make sure the app ***iSentryWebSettings.exe*** is allowed through Windows firewall.

License

Sentry Firefly on windows can be licensed by **Local License** or **Sentry LLS** Server. If you run **Sentry Firefly** and you see a pink icon you must license it.



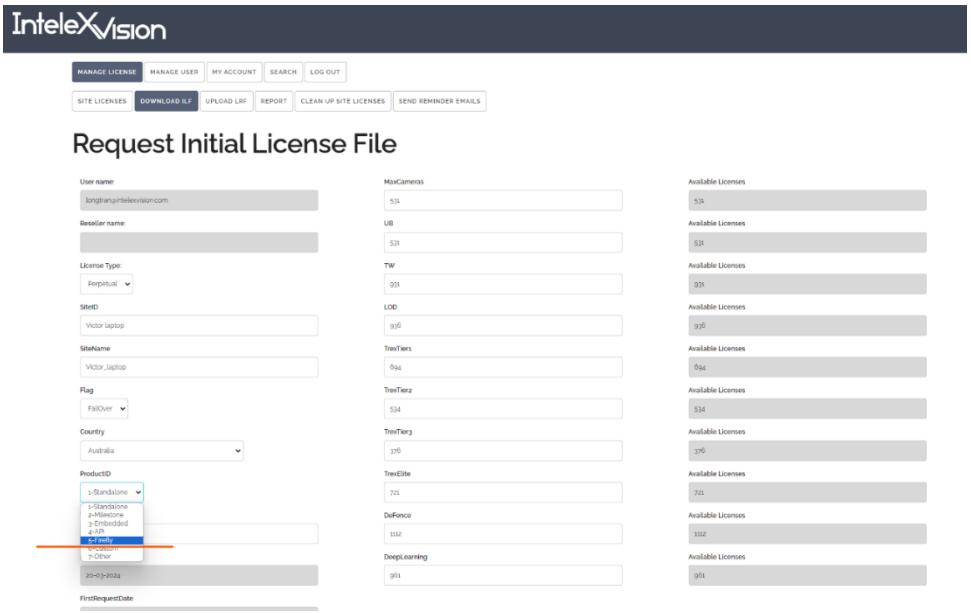
Please check relevant documents for details on how to license Sentry solutions or follow the next steps that should help to get **Sentry Firefly** licensed.

Access IntelexVision License Portal (https://intelexvision.com.au) with your credentials.

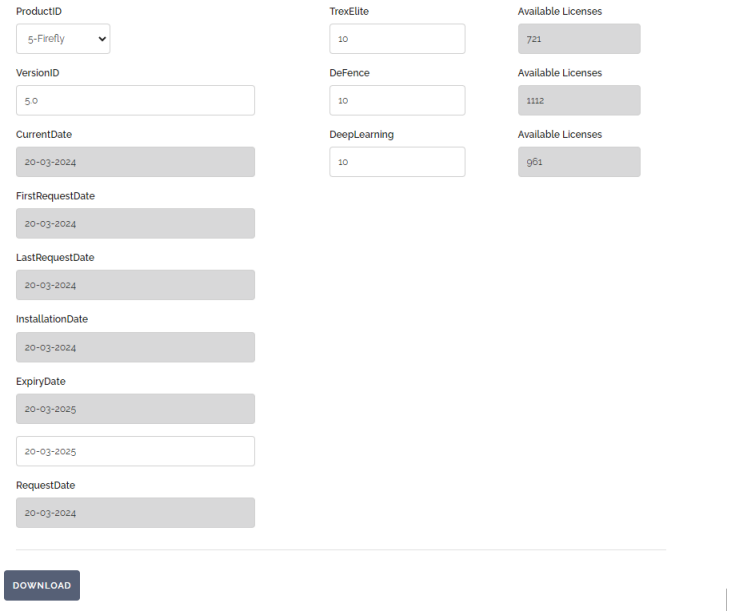
> **IMPORTANT NOTE**
>
> To license **Sentry Firefly** through **Sentry LLS**, the last one needs to be active with the correct licenses. Please check **Sentry LLS** guides for more information.

If the licensing method chosen is by **Local Licenses**, please follow the next steps.

1. Fill in all the detail for the license and choose Product ID equals to “5. Firefly”.

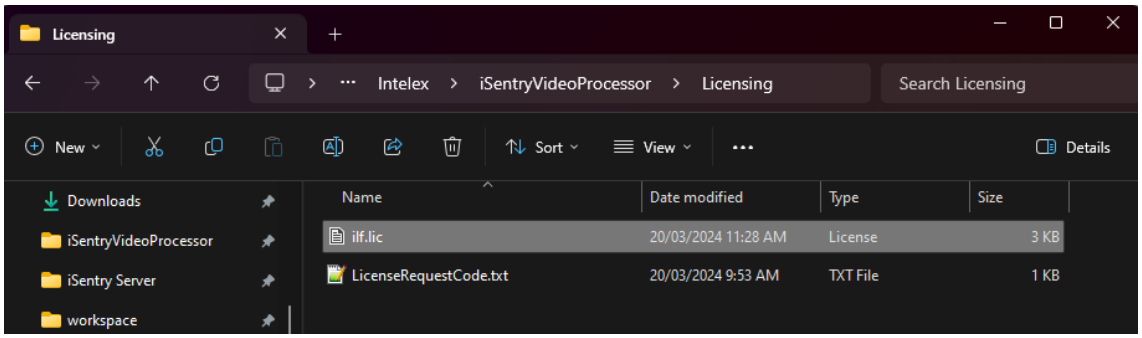


2. Download the file **ilf.lic** from the Portal by pressing “Download” button.

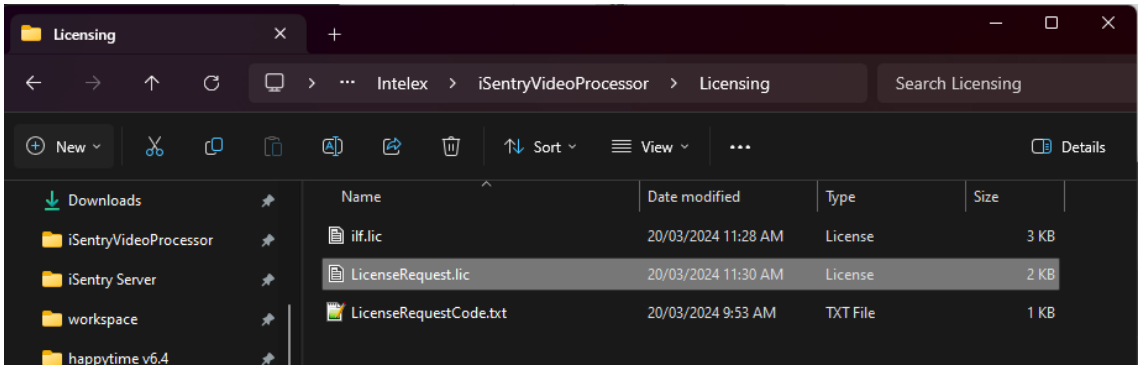


3. Place the file into default folder for Sentry LLS Server:

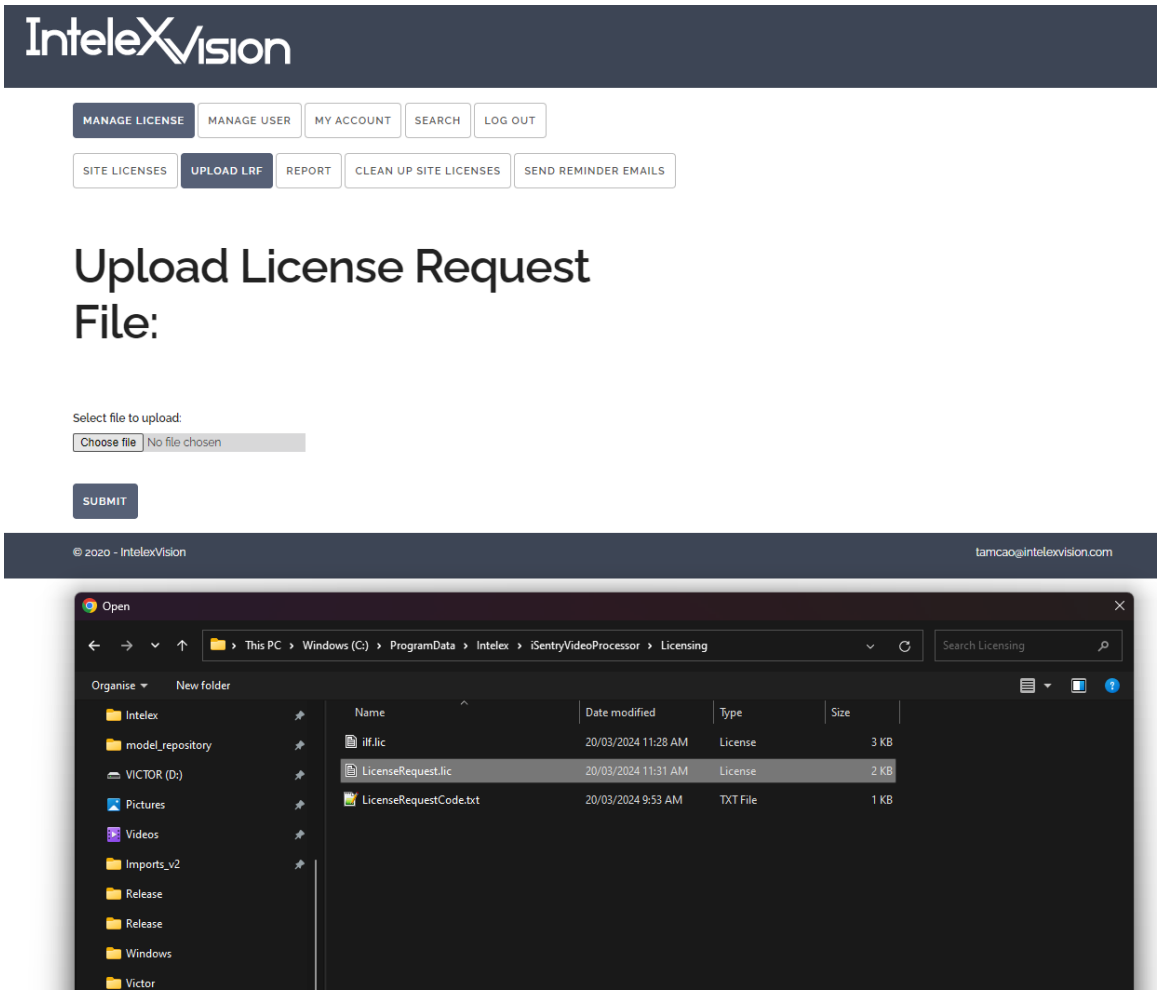
`C:\ProgramData\Intelex\iSentryVideoProcessor\Licensing`



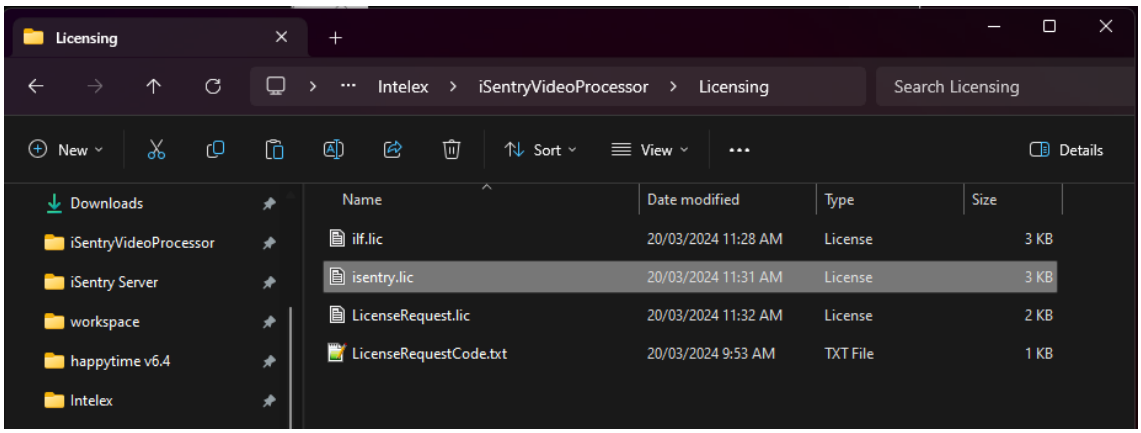
4. Then start **Sentry Firefly** from the *Tray Icon* and wait for the **LicenseRequest.lic** file to be generated.



5. This file should be uploaded to the Portal as detailed in the next image.



6. After pressing “Submit” button the **isentry.lic** (license) is downloaded and we can then move this file into **C:\ProgramData\Intelex\iSentryVideoProcessor\Licensing** to complete the licensing process.



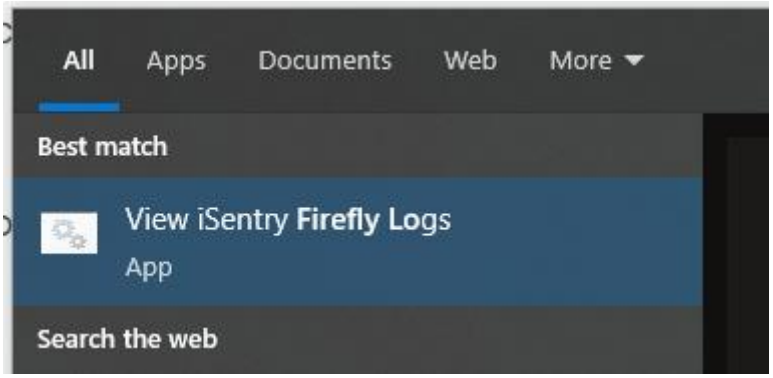
Configuration

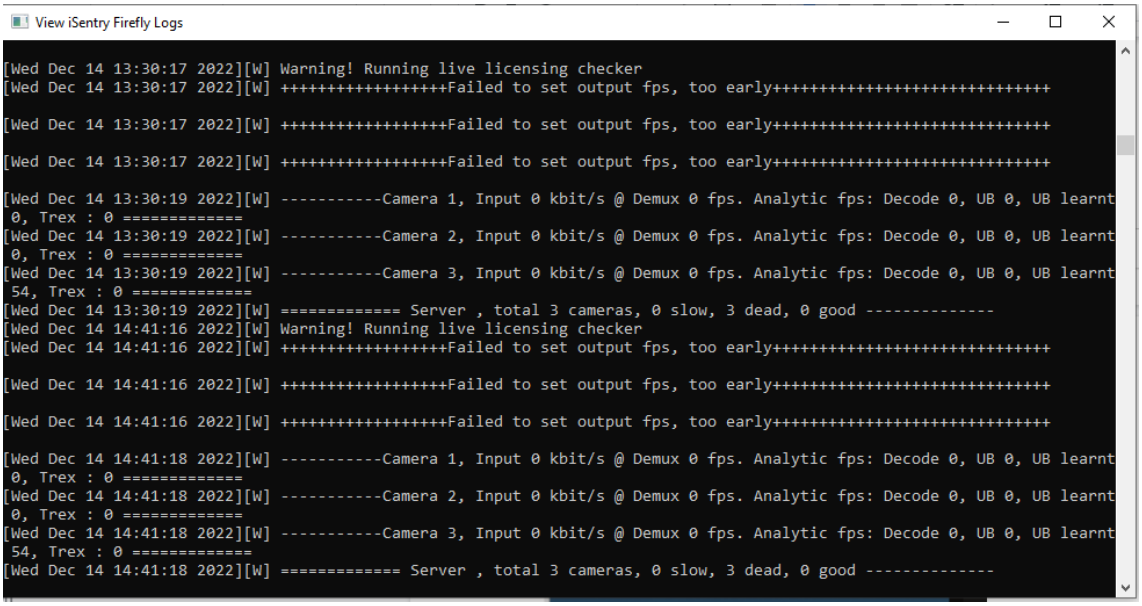
Access the ***Sentry Web Settings*** for configuring:

- Sentry LLS (Live Licensing Server).
- Camera Addition and Configuration.
- Video Analytic configuration.
- Integration with Clients to thread Alerts like WebClient, between others.

Check log file

There is a “live” console log equivalent that can be used to monitor or debug **Sentry Firefly** without the need to run it as console. To view it, from Start Menu, search for ***View iSentry Firefly logs*** and hit enter or click on the appropriate result.





Linux Installation

The installation in Linux systems lacks UI hence we have developed a docker release steps to make the experience smooth.

> **IMPORTANT NOTE**
>
> Docker needs to be properly installed prior to any further step.

Download images via docker hub repo

Sentry Firefly is packed under docker image.

Firefly images names list:

- Firefly image: **isentryintelex/isentry:firefly-2026R1**
- Web API/Settings: **isentryintelex/isentry:webserver_2026R1**
- Health Monitor: **isentryintelex/isentry:autoheal**

> **IMPORTANT NOTE**
>
> Following sample scripts are shown to illustrate the process but IntelexVision strongly recommend downloading the software resources from the portal.

Sample docker-compose script.yml:

```
volumes:
   models:

services:
   http_file_server:
      image: ${API_VER} # double check version using docker images command
      healthcheck:
             test:  curl  --fail  http://localhost:8123/  ||  curl  -k  --fail
https://localhost:8123 || exit 1
         interval: 1m
      volumes:
         - ${ISENTRY_PATH}:/usr/src/app/Intelex # CHANGE THE [ISENTRY_PATH] TO A
REAL PATH
         - "/etc/timezone:/etc/timezone:ro"
         - "/etc/localtime:/etc/localtime:ro"
         #- ./certs:/app/sslCerts
      ports:
         - 8123:8123
         #- 8124:8124 # uncomment if enable IV_ENABLED_WEB_LOGIN
      #entrypoint: /bin/bash
      environment:
         - IV_ENABLE_HTTPS_API=${IV_ENABLE_HTTPS_API}
         - ENABLED_WEB_LOGIN=${IV_ENABLED_WEB_LOGIN}
      restart: unless-stopped
   autoheal:
      restart: always
      image: willfarrell/autoheal:1.2.0
      environment:
         - AUTOHEAL_CONTAINER_LABEL=all
      volumes:
         - /var/run/docker.sock:/var/run/docker.sock

   firefly_models_default:
      image: ${FIREFLY_MODEL_base}
      container_name: firefly_models_default
      volumes:
         - models:/app/models
   firefly_models:
      image: ${FIREFLY_MODEL}
      container_name: firefly_models
      depends_on:
         firefly_models_default:
            condition: service_completed_successfully
      volumes:
```

```
        - models:/app/models

   iSentryFirefly_1:
      image: ${FIREFLY_VER} # double check version using docker images command
      container_name: iSentryFirefly_1
      depends_on:
         firefly_models:
            condition: service_completed_successfully
      environment:
         - SERVERID=ServerID_1
         - IV_ENABLE_HTTPS_API=${IV_ENABLE_HTTPS_API}
         - NVIDIA_VISIBLE_DEVICES=all
         - STATUS_ENDPOINT_HOST=http_file_server
      volumes:
         - ${ISENTRY_PATH}/ServerID_1:/root/Intelex
         - /dev:/dev
         - "/etc/timezone:/etc/timezone:ro"
         - "/etc/localtime:/etc/localtime:ro"
         - models:/app/models
      restart: unless-stopped
      cap_add:
         - SYS_PTRACE
         - SYS_RAWIO
      privileged: true
      logging:
         options:
            max-size: "501m"
            max-file: "10"
               entrypoint:    ./runVP.sh    "${IV_DETECTOR}"    "${MO_DL_MODE}"
${ENABLED_MODEL_OPTIMIZER:-}
      tty: true
      stdin_open: true
      runtime: ${DOCKER_RUNTIME:-runc}
```

Example of config.txt file

The lines starting with # are comments

```
#########################################################################
# IMPORTANT! ISENTRY_PATH TO SAVE THE SETTINGS FILE and LEARNING DATA.
ISENTRY_PATH=/home/user/Intelex
#########################################################################

######### set docker runtime, commented out if not using nvidia ##########
DOCKER_RUNTIME=nvidia
#########################################################################

# FIREFLY VERSION
FIREFLY_VER=comintelexvision/firefly-server:2026R1

FIREFLY_MODEL_base=comintelexvision/sentry-model:default-2026R1

# FIREFLY_MODEL
# choose one of those
# comintelexvision/sentry-model:openvino-2026R1 for openvino
# comintelexvision/sentry-model:nv-2026R1 for nvidia
# nvidia is default
FIREFLY_MODEL=comintelexvision/sentry-model:nv-2026R1


# WEB SETTINGS VERSION
API_VER=comintelexvision/firefly-websettings:2026R1
```

```
# SOME EXTRA FLAG 0: DISABLE, 1: ENABLED

## ENABLE HTTPS MODE
IV_ENABLE_HTTPS_API=0

# ENABLE WEB SETTINGS LOGIN. IF ENABLED, MAKE SURE PORT 8124 IS ENABLED IN DOCKER-
COMPOSE.YML FILE
IV_ENABLED_WEB_LOGIN=0

#      RUN      MODEL      OPTIMIZER      ONLY      -      REQUIRE      FOR      NVIDIA
==================================================

#ENABLED_MODEL_OPTIMIZER=1

## use keyword to add model
IV_DETECTOR="objcl ppe weapon-pose fire fall fighting"

# OBJECT_DETECTION_COLOR:     objcl
# FACE_DETECTION:             face
# HELMET_DETECTION:           helmet
# COLOR_CLASSIFICATION:       carcolor
# FIRE_N_THINGS_DETECTION:    fire
# FACE_MASK_COMPLIANCE:       facemask
# ALL_IN_ONE:                 allinone
# OBJECT_DETECTION_GRAYSCALE: objcl.gray
# POSE_VERIFICATION:          pose
# ACTION_RECOGNITION:         action
# PPE_DETECTION:              ppe
# WHEELCHAIR_DETECTION:       wheelchair
# HOSPITAL_BED_DETECTION:     hospbed
# WEAPON_DETECTION:           weapon
# ONE_HAND_RAISED:            handraised
# OBJECT_DETECTION_TOPDOWN:   topdown
# CROWD:                      crowd
# FIGHTING_DETECTION:         fighting
# FALL_DETECTION:             fall
# WEAPON_POSE_DETECTION       weapon-pose


# list used deep learning mode here, separated by a space " ". bal: BALANCE, acc:
ACCURACY, sft: SUPERFAST, spd: SPEED, hacc: high accuracy
MO_DL_MODE="bal spd"

#=============================================================================
================
```

Single Instance

> **INFORMATION**
>
> It is essential that the customer's is familiar with both Docker and CUDA, as this guide assumes those components are already properly installed and configured on the system.

- Edit *config.txt* file to changes:
    - iSentry Path
    - Firefly image name
    - Web Settings docker image name
    - HTTP/HTTPS option
    - User Login option
    - If using model optimizer, edit:
    - IV_DETECTOR to add/remove detection models. E.g. "objcl fire" will tell model optimizer to generate object detection and fire detection model
    - MO_DL_MODE to add/remove detection mode. E.g. "bal spd" will tell model optimizer to generate balance models and speed models
- To run model optimizer, execute command below:

```
ENABLED_MODEL_OPTIMIZER=1 docker compose --env-file config.txt up
```

- To run Firefly, execute command below:

```
docker compose --env-file config.txt up -d
```

- To stop Firefly, run command below:

```
docker compose --env-file config.txt down
```

Multiple Instances

- Check docker-compose-multi-instances.yml file for content of multiple instance script.
- Run command:

```
docker compose --env-file config.txt -f docker-compose-multi-instances.yml up -d
```

- Stop command:

```
docker compose --env-file config.txt -f docker-compose-multi-instances.yml down
```

> **INFORMATION**
>
> Accessing ***WebSettings*** will require extra param?ServerID=xx, e.g.:
>
> http://192.168.0.123:8123/settings?ServerID=1.

Tips

Simplify docker command by rename *config.txt* to *.env*.

You can simplify docker command FROM:

```
docker compose --env-file config.txt up -d
```

TO:

```
docker compose up -d
```

To rename config.txt to .env do move command:

```
mv config.txt .env
```

Troubleshooting and Common Issues

Reclaiming disk space

ModelCache

Sentry Firefly maintains an on-disk cache of generated models, which consumes disk space but can be reclaimed if needed. We recommend keeping the `ModelsCache` to allow users to switch between different Deep Learning model versions. However, if disk space is required, the cache can be safely deleted.

The path is here in Windows `C:\ProgramData\Intelex`

And here on Linux:

```
sentry@MSI:/home/user/Intelex/ServerID_1$ tree -d -L 2
.
├── ModelCache
│   ├── 1fc01194
│   ├── 8d0ab569
│   ├── a30ba12c
│   └── eed41f8b
└── iSentryVideoProcessor
    ├── AlertOutput
    ├── CameraData
    ├── Licensing
    ├── Logs
    ├── SettingsBackups
    └── images
```

Dump folder

When ***Sentry Firefly*** crashes, we keep a dump file which can take significant space in disk, if you need to reclaim space you can delete the files.

In Linux you can find them here: `/var/lib/systemd/coredump/`

In Windows in: `C:\ProgramData\Intelex\ iSentryVideoProcessor\dumps`

Support Information

If you need Technical Support with Intelex Vision systems, please write an email to customerservices@intelexvision.com and we will certainly help to solve the problem.