

# Adaptive Vision Library 4.10

---

## Introduction

Created: 24.08.2018

Product version: 4.10.2.62669

Table of content:

- SDK Installation
- Project Configuration

# SDK Installation

## Requirements

Adaptive Vision Library is designed to be a part of applications working under control of the Microsoft Windows operating system. Supported versions are: Vista, 7 and 8, as well as the corresponding embedded editions.

To build an application using Adaptive Vision Library, Microsoft Visual Studio environment is required. Supported versions are: 2010, 2012, 2013 and 2015.

Adaptive Vision Library can be also used on Linux operating system with GCC compiler - for details consult [Using SDK on Linux](#) article.

## Running the Installer

The installation process is required to copy the files to the proper folders and to set the environment variables used for building applications using Adaptive Vision Library.

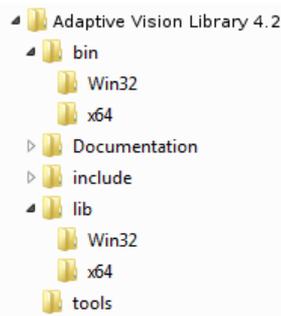
After the installation, a license for Adaptive Vision Library product has to be loaded. It can be done with the *License Manager* tool available in the *Start Menu*.

To verify that the installation has been successful and the license works correctly, one can try to load, build and run example programs, which are available from the *Start Menu*.

## SDK Directories

Adaptive Vision Library is distributed as a set of header files (.h), dynamic (.dll) and static (.lib) libraries. The libraries (static and dynamic) are provided in versions for 32-bit and 64-bit system. The header files are common for both versions.

The picture below shows the structure of the directories containing headers and libraries included in Adaptive Vision Library.



The directories (installed in the *Program Files* system folder) being a part of Adaptive Vision Library are shortly described below.

- **bin** – a directory containing dynamic linked library files (AVL.dll) for 32|64-bit applications. The libraries are common for all supported versions of Visual Studio and for Debug|Release configurations. All the functions of Adaptive Vision Library are included in the AVL.dll file.
- **Documentation** – a directory containing the documentation of Adaptive Vision Library, including this document.
- **include** – a directory containing all header (.h) files for Adaptive Vision Library. Every source code file that uses Adaptive Vision Library needs the AVL.h header file (the main header file) to be included.
- **lib** – a directory containing static (.lib) libraries ( AVL.lib ) for 32|64-bit applications. The AVL.lib file has to be statically-linked into the program that uses Adaptive Vision Library. It acts as an intermediary between the usage of Adaptive Vision Library functions and the AVL.dll file. The programmer creating an application does not need to bother about DLL entry points and functions exported from the AVL.dll file. Adaptive Vision Library is designed to be easy to use, so one only needs to link the AVL.lib file and can use all the functions from the AVL.dll just as easy as local functions.
- **tools** – a directory containing the *License Manager* tool helping the user to load the license for Adaptive Vision Library to the developer's computer.
- **Examples** – a directory located in the *Public Documents* system folder (e.g. *C:\Users\Public\Documents\Adaptive Vision Library 4.10\Examples* on Windows Vista/7) containing simple example solutions using Adaptive Vision Library. The examples are a good way of learning, how to use Adaptive Vision Library. They can be used as a base for more complicated programs as well. The shortcut to the Examples directory can be found in the *Start Menu* after the installation of Adaptive Vision Library.

## Environment and Paths

Adaptive Vision Library uses the environment variable named *AVL\_PATH410* (410 stands for the 4.10 version) in the building process. The variable points the directory with the headers and libraries needed in the compile time (.h files and AVL.lib ) and in the run time ( AVL.dll ). Its value is typically set to "*C:\Program Files (x86)\Adaptive Vision\Adaptive Vision Library 4.10*", but it can differ in other systems.

The projects using Adaptive Vision Library should use the value of *AVL\_PATH410* to resolve the locations of the header files and statically-linked AVL.lib file. Using an environment variable containing path makes the application source code more portable between computers. The *AVL\_PATH410* path is typically used in the project settings of the compiler (Configuration Properties | C/C++ | General | Additional Include Directories) to find the header files, settings of the linker (Configuration Properties | Linker | General | Additional Library Directories) to find the proper version of the AVL.lib and in the configuration of Post-Build Event (Configuration Properties | Build Events | Post-Build Event | Command Line) to copy the proper version of the AVL.dll file to the output directory of the project. All the settings can be viewed in the simple example applications distributed with Adaptive Vision Library.

# Project Configuration

## General Information

Adaptive Vision Library is designed to be used as a part of C++ projects developed with Microsoft Visual Studio in versions 2010-2015.

## Creating a New Project

### Microsoft Visual Studio 2010

Adaptive Vision Library is provided with a wizard that creates and configures a project for using Adaptive Vision Library. To use the wizard, start Microsoft Visual Studio and choose the *File | New | Project...* command. The wizard called *AVLProject* is available at *Installed Templates | Visual C++ | AVL*. Enter the name for your project and click OK. An empty project configured for using Adaptive Vision Library will be created.

### Microsoft Visual Studio 2012, 2013 and 2015

A project template similar to the wizard available for Visual Studio 2010 is available also for the newer versions. To create a new project using Adaptive Vision Library, start Microsoft Visual Studio and choose the *File | New | Project...* command. The template called *AVLProject* is available in the tab *Installed | Templates | Other Languages | Visual C++*.

## Required Project Settings

All projects that use Adaptive Vision Library need some specific values of the compiler and linker settings. If you want to use the Library in your existing project or you are manually configuring a new project, please apply the settings listed below:

- **Configuration Properties | General**
  - **Character Set** should be set to *Use Unicode Character Set*.
- **Configuration Properties | C/C++**
  - **General**
    - **Additional Include Directories** should contain the `$(AVL_PATH410)\include\` path.
- **Configuration Properties | Linker**
  - **General**
    - **Additional Library Directories** should contain the proper path to directory containing the `AVL.lib` file. The proper path is `$(AVL_PATH410)\lib\$(PlatformName)\`.
  - **Input**
    - **Additional Dependencies** should contain `AVL.lib` file.
- **Configuration Properties | Build Events**
  - **Post-Build Event**
    - **Command Line** should contain `copy "$(AVL_PATH410)\bin\$(PlatformName)\AVL.dll" "$(OutDir)"` call. This setting is not mandatory, but the application using Adaptive Vision Library requires an access to the `AVL.dll` file and this is the easiest way to fulfill this requirement.

## Including Headers

Every source code file that uses Adaptive Vision Library needs the `#include <AVL.h>` directive. A proper path to the `AVL.h` file is set in the settings of the compiler (described above), so there is no need to use the full path in the directive.

## Distributing Adaptive Vision Library with Your Application

Once the application is ready, it is time for preparing a distribution package or an installer. There are two requirements that needs to be fulfilled:

- The final executable file of the application needs to have access to the proper version (used by *Win32* or *x64* configuration) of the `AVL.dll` file. Typically, the `AVL.dll` file should be placed in the same directory as the executable.
- The computer that the application will run on needs a valid license for the use of Adaptive Vision Library product. Licenses can be managed with the License Manager application, that is installed with Adaptive Vision Library Runtime package.
- A license file (\*.key) can be also manually copied to the end user's machine without installing Adaptive Vision Library Runtime. It must be placed in a subdirectory of the *AppData* system folder. The typical location for the license file is `C:\Users\  
<UserName>\AppData\Local\Adaptive Vision\Licenses`. Remember that the license is valid per machine, so every computer that runs the application needs a separate license file.

# Using Library on Linux

## Requirements

Adaptive Vision Library is designed to be used with GCC compiler on Linux x86\_64, embedded ARMv7-A and ARMv8IA systems. Currently gcc in version 5.4 is supported, and corresponding toolchains for embedded linux: arm-linux-gnueabihf-, aarch64-linux-gnu-. Custom build can be prepared upon the earlier contact with Adaptive Vision team. The Adaptive Vision Library is distributed as .tar.gz archive. The library is compatible with Debian-like system, including - but not limited to - Ubuntu distributions.

## Common prerequisites

Properly set locale on target computer is important. Non-existing locale will cause bugs and bad behavior. To list locale that exists on your computer use: `locale -a`, and currently set: `locale`. Remember that running your application as daemon from ie. systemd may set different locale, than the one in your user terminal. Refer to your Linux distribution documentation.

Libraries needed to link programs:

- libopenmp
- libpthread
- librt
- libdl

Runtime libraries for libavl:

- libudev

For some examples to build, additional libraries listed below may be needed:

- libgtk-3-dev / libgtk2.0-dev
- libsdl-dev
- qt5

To build example in simple manner, GNU Make tool is needed.

To install all development requirements (including examples):

```
sudo apt-get install libudev1 g++ make libgtk-3-dev libsdl-dev qtbase5-dev
```

To install only libavl development requirements:

```
sudo apt-get install libudev1 g++ make
```

## Compilation instructions

### Directory structure

Unpacked directory consists of following entries:

- examples/ - directory contains source files of example programs written with Adaptive Vision Library
- include/ - this directory contains library header files
- lib/ - here the .so file with library is stored, along with any kits
- bin/ - directory for additional binaries, like Licensing tool.
- /README - instruction of library usage
- /sha512sum - checksums for all files in archive, check with `sha512sum --quiet -c sha512sum`
- /system - file containing information about the optimal target system - contains lines for OS, architecture, specific optimizations

### Compilation

For compiling with Adaptive Vision Library please remember to:

- add the include/ subdirectory to the compiler include directories: `-I` switch
- add the lib/ subdirectory to the linker directories: `-L` switch
- link with Adaptive Vision Library: `-lavl` switch
- use `-rpath` in linker options, `LD_LIBRARY_PATH` or `LD_PRELOAD` of `libavl.so` file.
- link with dependencies: `-fopenmp -lpthread -lrt -ldl`

One can consult makefile in the examples/ directory to see how to compile and link with Adaptive Vision Library.

# Licensing and distribution

## Licensing

File based licenses are supported on all Linux platforms. Dongle licenses depend on CodeMeter runtime. Currently codemeter runtime is available for x86\_64 and ARMV7-A. To develop and debug programs written with Adaptive Vision Library, Library license has to be present. To run compiled binaries linked with Adaptive Vision Library, LibraryRuntime license has to be present.

One can use `license_manager` from `bin/` directory to list currently installed file or dongle licenses: `license_manager -l`

### File License

To obtain license:

- In a terminal, on the target machine run `license_manager --id` from `bin/` directory
- Copy the printed Computer ID
- Use that Computer ID to get a `.key` file from User Area on [www.adaptive-vision.com](http://www.adaptive-vision.com) website.
- Download the key to the target machine
- Run in terminal `license_manager -i downloaded_file.key`

### Dongle License

Installed CodeMeter Runtime is required, as well as proper license available on plugged in dongle.

Download runtime package from [WIBU website](http://www.adaptive-vision.com), section "CodeMeter User Runtime for Linux".

"Driver Only" (lite) version recommended for headless (no desktop GUI) installations. ARMV7-A is available under "CodeMeter User Additional Downloads" as "Raspberry PI" version

## Distribution

To distribute program with Adaptive Vision Library, one have to provide license (file or dongle - depending on system used) and the `libavl.so`. To provide the `.so` file, one can install SDK on target machine, but this will provide headers etc., which may be unwanted. In such case, the library file, with any used kits should be copied to suitable system directory, or the program has to be compiled with `-rpath` and relative path to the `.so` file. Third option is to provide a boot script, which will set `LD_LIBRARY_PATH` or `LD_PRELOAD` with `libavl.so` location.

## Program development - general advise

The most convenient way to make programs with Adaptive Vision Library for Linux is to develop vision algorithm using Adaptive Vision Studio on Windows and then generating C++ code. This code can be further changed or interfaced with rest of the system and tested on Windows. Then, cross-compiler can be used to prepare Linux build, which will be provided to target machine. It is easy to organize work this way, because:

- developing vision algorithm using plain C++ is hard, troublesome and error prone, but Adaptive Vision Studio makes it easy,
- programs written with Adaptive Vision Library on Windows can be easily debugged using Visual Studio thanks to provided debug visualisers and the Image Watch extensions to Visual Studio,
- cross compilation using virtualization solution, like Vagrant, is easy and fast, and does not force developer to use two systems simultaneously.

Of course, the programs can be also developed on Linux machine directly. Then a dose of work should be put into writing good Makefile. Debugging can be done by GDB, but we do not provide debug symbols for Adaptive Vision Library.



This article is valid for version 4.10.2

©2007-2018 Future Processing