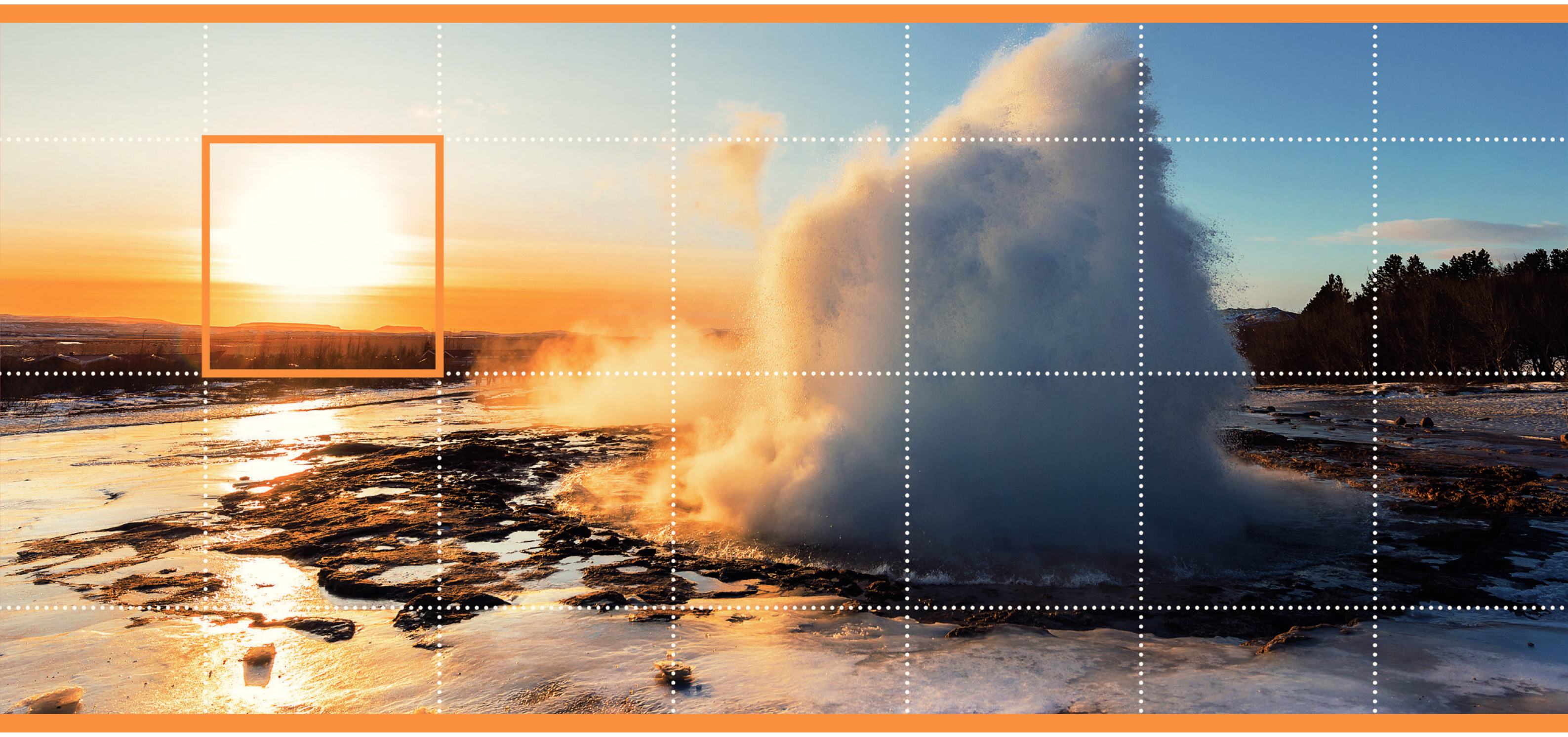
HALCON

a product of MVTec

Installation Guide



HALCON 17.12 *Progress*

MVTec Software GmbH

Building Vision for Business

All about installing and licensing HALCON, Version 17.12

More information about HALCON can be found at: `http://www.halcon.com/`

About This Manual

The manual provides the necessary information to install HALCON and setup the licensing mechanism successfully. It is divided into the following chapters:

- **Introduction**
  A short overview of the different HALCON versions, available licensing schemes, and the system requirements.
- **Installing HALCON**
  How to install HALCON, either for the first time or in form of an update or upgrade.
- **All About HALCON Licenses**
  Detailed information about the different types of licenses and how to obtain and install them.
- **Troubleshooting**
  Possible problems and how to solve them.
- **Appendix**
  Details like the installed file structure and the relevant environment variables.

For **further information** about HALCON, please consult the Quick Guide, section 1.5 on page 7, which gives an overview of the available documentation. ☜

Notation

Except for Linux/macOS-specific sections, file paths and environment variables are printed in the Windows convention, e.g.,

```
%HALCONEXAMPLES%\extension_package\halconuser
```

to denote the subdirectory `halconuser` containing an example package within the HALCON examples directory referenced by the environment variable `HALCONEXAMPLES` (see section A.2 on page 42 for more information on environment variables). The same expression in Linux/macOS convention would look like

```
$HALCONEXAMPLES/extension_package/halconuser
```

Contents

1 Introduction 7

1.1 HALCON Configurations . . . 7

1.2 Releases and HALCON Versions . . . 7

1.3 Supported Platforms and Minimum System Requirements . . . 8

1.3.1 Platform-Specific HALCON Versions . . . 8

1.3.2 Platform-Independent Applications . . . 9

1.3.3 HALCON Variable Inspect (Visual Studio Extension) . . . 9

1.3.4 Requirements for Deep Learning . . . 9

1.4 Licensing . . . 10

2 Installing HALCON 11

2.1 How to Obtain HALCON . . . 11

2.2 Installing HALCON for the First Time . . . 11

2.2.1 Windows . . . 12

2.2.2 Linux . . . 14

2.2.3 macOS . . . 16

2.3 Installing Additional Parts of HALCON . . . 17

2.4 Manual Runtime Installation . . . 17

2.5 Updating HALCON . . . 18

2.6 Upgrading HALCON . . . 18

2.7 HALCON Installer Switches under Windows . . . 18

2.8 Uninstalling HALCON . . . 19

2.8.1 Windows . . . 19

2.8.2 Linux . . . 20

2.8.3 macOS . . . 20

2.9 Managing Multiple HALCON Versions . . . 21

2.10 Switching between Different HALCON Platform Versions under Windows . . . 21

2.11 Installing HALCON I/O Device and Image Acquisition Interfaces . . . 21

2.12 Installing HALCON Extension Packages . . . 22

2.12.1 Using an Extension Package Within HDevelop . . . 22

2.12.2 Using an Extension Package in a Stand-Alone Application . . . 22

3 Installing Third-Party Components 25

3.1 Getting the Components . . . 25

3.2 Copying the Libraries . . . 25

4 All About HALCON Licenses 27

4.1 What is a License? . . . 27

4.2 Evaluation Licenses . . . 28

4.3 Development Licenses . . . 29

4.3.1 License Bound to a Network Card . . . 29

4.3.2 License Bound to a Dongle . . . 29

4.4 Runtime Licenses . . . 30

4.5 How to Upgrade a License . . . 31

4.6 How to Check the Genuineness of HALCON (Windows) . . . 31

5 Troubleshooting 33

5.1 Problems During Installation (Windows) . . . 33

5.2 Problems During Uninstallation (Windows) . . . 33
5.3 Problems Concerning Licenses . . . 34
5.3.1 Extracting Host IDs . . . 34
5.4 Troubleshooting in HDevelop or HALCON Applications . . . 35
5.4.1 Startup Errors . . . 35

A More on the Installation **37**
A.1 The Installed File Structure . . . 37
A.1.1 Main Directory . . . 37
A.1.2 Example Programs, Images, Models . . . 41
A.1.3 Machine Configuration Data . . . 42
A.1.4 User Configuration Data . . . 42
A.2 HALCON's Environment Variables . . . 42
A.2.1 Setting Environment Variables Under Windows . . . 42
A.2.2 Setting Environment Variables Under Linux . . . 43
A.2.3 HALCON-Specific Environment Variables . . . 43
A.2.4 General Environment Variables . . . 44
A.3 Registry Keys on Windows Sytems . . . 44
A.3.1 Registered File Types . . . 44
A.3.2 Environment Variables . . . 44
A.3.3 Basic Installation Data . . . 44
A.3.4 Uninstall Information . . . 45

Index **47**

Chapter 1

Introduction

To use HALCON on a computer, you must

1. install HALCON on this computer and
2. obtain a license.

Before looking into the details of these two steps in chapter 2 on page 11 and chapter 4 on page 27, this chapter gives an overview of the different HALCON versions and licensing methods. Finally, it describes the system requirements for running HALCON.

1.1 HALCON Configurations

You can use HALCON in two configurations:

1. **Development version**
   The development version (sometimes also denoted as full version) includes the full spectrum of HALCON, i.e., language interfaces to C, C++, and .NET, interfaces to image acquisition and I/O devices, the Extension Package Interface, which allows you to integrate your own HALCON operators, and, of course, the interactive development environment HDevelop. You need this version whenever you want to develop applications based on HALCON.

2. **Runtime version**
   If you have finished developing an application based on HALCON, you only need a runtime version of HALCON for each computer where the application is to be run. Since the runtime version is not determined for developing applications it does not include the interactive development environment HDevelop. Furthermore, you can obtain runtime versions that include only parts of the functionality (so-called *modules*); please contact your local distributor for more information.

1.2 Releases and HALCON Versions

The term *version* has a second meaning: It denotes the major HALCON releases, e.g., HALCON 17.12, to differentiate them from so-called *maintenance releases* like HALCON *x*.*y*.1. The main differences between these two types of releases are:

- **Functionality**
  A new HALCON *version* always represents a major step in the functionality. This means that it contains a significant number of new operators, but possibly also new functionality in HDevelop, e.g., new assistants. Furthermore, the functionality of individual operators may be extended or operators have been sped up. Of course, all currently known bugs in the preceding release will have been fixed.

  In contrast, the main intention of a *maintenance release* is to fix all currently known bugs. Nevertheless, such a release typically also brings some speed-ups and minor functional extensions.

- **Compatibility**
  A new HALCON *version* is not downward compatible, with the following implications: First, you must upgrade your HALCON license (see section 4.5 on page 31). Second, if you want to run applications created with an older release under the new version, you must regenerate the applications, as the new HALCON library is not binary compatible to the old one. The term 'applications' includes also image acquisition interfaces and extension packages you created yourself based on an older release. Note that a new version may also be source-code incompatible in some parts, e.g., the signature of an operator or a class method may have been changed. These changes are indicated in the release notes of the HALCON version. In such a case, you must adapt the source code of your application before regenerating it.

  In contrast, a *maintenance release* is in most cases fully downward compatible to its corresponding version. This compatibility includes the license. Please note, however, that some maintenance releases may not be fully binary or source-code compatible because of technical reasons. In such cases, the release notes will contain corresponding warnings and describe how to proceed.

1.3 Supported Platforms and Minimum System Requirements

HALCON runs on Windows, Linux, and macOS. The minimum system requirements are listed in table 1.1; more details follow below.

	Windows		**Linux**		**macOS**
	x86	x64	x64	armv7a	x64
Processor	Intel Pentium 4 or AMD Athlon 64 with SSE2	Intel 64 or AMD 64	Intel 64 or AMD 64	Armv7-A with NEON support	Intel 64
Disk Space (Full / Runtime)	2.0 GB / 340 MB				
Memory	256 MB				
Display Resolution	1024 × 768				
OS Versions	Windows 7, 8, 8.1, 10 or Windows Server 2008 R2, 2012, 2012 R2, 2016	Windows 7, 8, 8.1, 10 or Windows Server 2008 R2, 2012, 2012 R2, 2016 x64 Edition	Linux x86_64	Linux armv7a	macOS 10.12, 10.13
Compiler / Environment	Visual Studio 2005	Visual Studio 2005	gcc 4.8	gcc 4.8	Xcode
HALCONARCH	`x86sse2-win32`	`x64-win64`	`x64-linux`	`armv7a-linux`	`x64-macosx`
Specifics				Kernel with hidraw support, hard-float ABI	
Libraries			GLIBC_2.17, GLIBCXX_3.4.15	GLIBC_2.7, GLIBCXX_3.4.4	
Application-specific Requirements			X11R7, freetype 2.4.11, fontconfig 2.10.95, OpenGL 2.0	X11R7, freetype 2.4.11, fontconfig 2.10.95, OpenGL 2.0	
HALCON Variable Inspect	see section 1.3.3 on page 9				
Deep Learning	(not available)	see section 1.3.4 on page 9		(not available)	(not available)

Table 1.1: Minimum system requirements.

1.3.1 Platform-Specific HALCON Versions

For each of the operating systems listed in table 1.1 on page 8, platform-specific versions of HALCON's executables and libraries are provided. The name of the currently used version is stored in the environment variable `HALCONARCH`. On Windows and Linux, HALCON uses AVX2-optimized code for many operators, when run on a machine that supports AVX2.

Note that HALCON should also run on newer versions of the operating systems than the ones listed; however, we cannot guarantee this.

HALCONARCH appears in several directory paths: Executable HALCON programs like hdevelop, and DLLs like halcon.dll (Windows only), reside in %HALCONROOT%\bin\%HALCONARCH%. On Windows systems, this path is therefore automatically included in the environment variable PATH; on a Linux system, you must include it in your login script.

The libraries that you need for linking programs, e.g., halcon.lib (Windows) or libhalcon.so (Linux) reside in the directory %HALCONROOT%\lib\%HALCONARCH%. On macOS, the corresponding libraries are organized as frameworks in /Library/Frameworks.

Please note that when creating a 64-bit application, both the development computer and the computer on which the application will run must be 64-bit platforms. On the other hand, you can use a 32-bit HALCON version on a 64-bit platform using the WOW64 (Windows-On-Windows 64-bit) subsystem.

In order to create .NET applications under Linux, you need to install Mono.

1.3.2 Platform-Independent Applications

Even when using a platform-specific version of HALCON, you can still create platform-independent applications, in two ways:

- **With HDevelop**, HALCON's integrated development environment (IDE). HDevelop programs are stored in a platform-independent format, thus, you can run them on any supported platform.
- **With HALCON/.NET**, HALCON's interface to .NET programming languages. Applications written in .NET languages are stored in a platform-independent intermediate language, which is then converted by the so-called common language runtime into platform-specific code.

You can combine both methods by using HDevEngine/.NET to run HDevelop programs from a HALCON/.NET application.

1.3.3 HALCON Variable Inspect (Visual Studio Extension)

HALCON Variable Inspect depends on the following components:

- Visual Studio 2013 (Update 5 or higher) through Visual Studio 2017
- .NET 4.6

Older installations of Visual Studio 2013 might be missing the .NET 4.6 framework. If the installation of HALCON Variable Inspect fails with an error, install the .NET 4.6 framework first.

1.3.4 Requirements for Deep Learning

Deep Learning with Convolutional Neural Networks is implemented to run on NVIDIA GPUs and uses the libraries cuDNN and cuBLAS. Thereout result additional system requirements, listed in table 1.2. The use of the cuDNN library leads to the requirement of an operating system officially supported by NVIDIA and an appropriate graphics driver. Please note, **Windows 32-bit, macOS, and Linux armv7a** systems are not supported yet.
We recommend to check with NVIDIA, which operating system for Windows x64 and Linux x86_64 is currently supported. Information concerning the installation of the libraries cuDNN and cuBLAS can be found in the Installation Guide, chapter 3 on page 25

GPU	Operating System	NVIDIA Driver
NVIDIA GPU with compute capability 3.0 or higher	Windows x64 Linux x86_64	Supporting at least CUDA 8

Table 1.2: Additional system requirements for HALCON Deep Learning using CNN.

1.4 Licensing

To run HALCON on a computer, you need a license. Licenses are always issued for a certain HALCON version (i.e., major release, see section 1.2 on page 7). However, a license is not exclusively bound to this version: It is *upward compatible within the version number*, i.e., licenses for major releases are also valid for follow-up maintenance releases.

The license file is typically called "`license.dat`". Between "license" and ".dat", arbitrary information can be included. Thus, different licenses for different versions (e.g., `license-17.12.dat` for version 17.12) can be stored in the same directory.

The three possible licensing types mainly correspond to the different HALCON versions described in section 1.1 on page 7. Detailed information about HALCON licenses can be found in chapter 4 on page 27.

- **Evaluation license**
  To evaluate the full power of HALCON, you can obtain an evaluation license from your local distributor free of charge. This type of license is not bound to any computer hardware, i.e., you can use HALCON on any computer you installed it on; however, it is only valid for a limited time, typically for a month. Note that you may not use this license to develop commercial applications.

- **Development license**
  To develop HALCON applications, whether in HDevelop or via a programming language, you need a development license. In contrast to the evaluation license, this license is permanent. Furthermore, this license is bound to a certain hardware component (network card or dongle, see section 4.1 on page 27).

  If you want to use HALCON on multiple computers simultaneously, you need a license for each of them.

- **Runtime license**
  If you finished developing your application based on HALCON and now want to install and run it on a customer's computer, you only need a runtime license. Like development licenses, runtime licenses are permanent and bound to a certain hardware component (network card or dongle).

  As already noted, you can obtain runtime licenses that cover only parts of the functionality (so-called *modules*). Please contact your local distributor for more information.

Chapter 2

Installing HALCON

In this chapter, we show how to

- obtain HALCON (section 2.1 on page 11), and install it for the first time (section 2.2 on page 11),
- install additional parts (section 2.3 on page 17),
- manually install a runtime version of HALCON (section 2.4 on page 17),
- update HALCON to a newer maintenance release (section 2.5 on page 18), and
- upgrade HALCON to a newer version (section 2.6 on page 18).

Furthermore, this chapter explains how to

- uninstall HALCON (section 2.8 on page 19),
- manage multiple versions in parallel (section 2.9 on page 21),
- switch between HALCON for 32-bit and 64-bit Windows or vice versa (section 2.10 on page 21),
- install updates of image acquisition interfaces (section 2.11 on page 21), and
- install extension packages (section 2.12 on page 22).

2.1 How to Obtain HALCON

You can obtain HALCON by downloading it from `http://www.halcon.com/download`. Note that you must first register yourself before downloading software. The download page provides separate files for the different configurations (see section 1.1).

2.2 Installing HALCON for the First Time

In the following sections, we explain how to install HALCON for the first time

- on Windows systems: section 2.2.1
- on Linux systems: section 2.2.2 on page 14
- on macOS systems: section 2.2.3 on page 16

Administrator privileges are required to install HALCON under Windows and macOS, and also on Linux systems if a dongle driver is used.

2.2.1 Installing HALCON under Windows

- Assisted installation: section 2.2.1.1
- Silent runtime installation: section 2.2.1.2 on page 14

2.2.1.1 Assisted Installation (full or runtime version)

To install HALCON on Windows systems, simply double-click the downloaded installer.

The installer checks for previous installations of HALCON. Depending on the installed version (see section 1.2 on page 7), the installer will offer an appropriate action:

- If a different major version is installed, you can choose which version will be active after the installation.
- If a different minor version is installed, the installer will update the existing installation.

Step 1: Internal checks, license agreement, maintenance releases

First, the setup program checks whether your system meets the requirements for running HALCON (see section 1.3 on page 8). Besides, it checks whether you already installed HALCON on this computer. If not, a license agreement is displayed; its text is stored in the file `eula.txt`, which will be installed in the folder you select for the installation.

The installer also checks for available maintenance releases (full version only). You can skip this step, e.g., if the computer is not connected to the Internet. Afterwards, the most appropriate architecture for the installation is suggested (x86 or x64), if applicable.

Step 2: Select installation type

Then, you are asked to select between the following installation types which correspond to the HALCON configurations described in section 1.1 on page 7. Note that you can install additional parts of HALCON at a later time by starting the setup program again and selecting the parts you need (see section 2.3 on page 17). The runtime installer only supports the component selections “Runtime” and “Custom”.

Full

This installs the complete development version of HALCON, i.e., the full set of libraries and executables including interfaces to image acquisition devices and programming languages. Furthermore, it installs the full documentation, and the full set of example programs (see section A.1 on page 37 for an overview), including all necessary images and 3D models.

Compact

This installs a minimal development version of HALCON, i.e., besides the full set of libraries and executables including interfaces to image acquisition devices and programming languages the installation encompasses only the User’s Manuals and the Reference Manual. Note that neither examples nor images are installed.

Runtime

This installs the runtime version of HALCON, i.e., only the set of libraries including interfaces to image acquisition devices, I/O device interfaces, and programming languages that are necessary to run a HALCON application. Naturally, this type of installation encompasses neither documentation, examples nor images. However, it contains HDevEngine and the provided external procedures.

Custom

Like the `Full` and the `Compact` installation, this type installs the development version of HALCON. As its name suggests, it lets you select which parts of the documentation, examples, and images you want to be installed. Note that some HALCON experience is required to handle this type of installation.

Step 3: Additional drivers for GigE Vision

The installer offers to install the MVTec GigE Vision streaming filter driver. This option is recommended if you intend to acquire images from GigE Vision compliant cameras via the HALCON GigEVision interface because it increases the performance significantly.

Step 4: Documentation language (full version only)

By default, the documentation will be installed in English. The HALCON operator reference is also available in German and Japanese. You can select the desired language of this manual.

Step 5: Select installation destination

After these checks you can choose a folder where HALCON is to be installed, e.g., `C:\Program Files\MVTec\HALCON-17.12`.

Step 6: Example programs and images (full version only)

The installer will install example programs as shared user documents by default, i.e., in a location which is writable by all users. This is highly recommended because some of the example programs require write access to run properly. Alternatively, you can install these example programs inside the installation destination specified in the previous step. After the installation, the environment variable `%HALCONEXAMPLES%` points to the actual location of the example programs, and the environment variable `%HALCONIMAGES%` points to the location of the images.

Step 7: The actual installation

Finally, the actual installation starts, which includes copying files to the destination folder (see section A.1 on page 37 for an overview of HALCON's fileset), entering information in the Windows registry, and setting environment variables (see also section A.2 on page 42). Depending on the installation type and content, it may be necessary to reboot the computer after the installation is finished.

Step 8: HALCON Variable Inspect

HALCON includes an extension that simplifies the debugging of applications in Visual Studio. During the installation, the installer checks for a compatible version of Visual Studio (see section 1.3.3 on page 9) and offers to install the extension. If you wish to install the extension later, double-click `%HALCONROOT%\misc\HALCON_Variable_Inspect_1712.vsix` and follow the instructions. See the Programmer's Guide, section 3.3 on page 24 for more information about HALCON Variable Inspect.

If the installation of the **Variable Inspect extension for Visual Studio** fails, please update your Visual Studio 2013 to Update 5 or higher.

Step 9: License installation (optional)

After the setup of HALCON has finished, the installer lets you install a license file. This step can be skipped if you have not yet obtained a license from your local distributor. To install the license at a later time, simply copy the obtained license file `license.dat` (or similar) to the directory `%HALCONROOT%\license`.

Apart from that, no further action on your part is required. You can immediately use HALCON, e.g., experiment with HDevelop if you have obtained a license already. A good starting point is the Windows start menu, via which you can read the manuals or the release notes, or start HDevelop.

If you are using a **CodeMeter dongle and CodeMeter Runtime** is running, CodeMeter Runtime needs to be at least version 5.00. Otherwise, the dongle is not recognized correctly, and HALCON throws an error regarding the license file.

Step 10: Further configuration

Optionally, you can optimize **HALCON's automatic operator parallelization** to your computer as described in the Programmer's Guide, section 2.1.1 on page 15.

Step 11: Additional steps for Deep Learning

In order to use the Deep Learning functionality, two third-party libraries need to be installed manually (see chapter 3 on page 25).

Futhermore, the example programs for Deep Learning rely on the installation of a set of images. These images are available in a separate installer `halcon-17.12.0.0-windows-images-deep-learning.exe`, which can be downloaded from `http://www.halcon.com/download`.

2.2.1.2 Silent Installation (runtime version only)

This installation mode is provided for automatic, unattended installations of the runtime version of HALCON. Open a Windows command prompt (as Administrator), and change to the directory where you put the downloaded runtime installer.

Start the silent runtime installation using the /S switch:

```
halcon-17.12.0.0-windows-runtime.exe /S
```

In this mode, no warnings are given when encountering existing installations:

- Runtime binaries of existing HALCON 17.12 installations will be overwritten.
- Other versions will not be uninstalled.

The silent runtime installation supports the following additional command line parameters (only for first-time installations, not in the case of an update):

- /DEST="target directory"
  Default: %PROGRAMFILES%\MVTec\HALCON-17.12.
- /ARCH=[x86sse2-win32|x64-win64]
  Default: Best supported architecture is installed.
- /DRIVER=gigevision
  Default: No driver is installed.
- /UNSELECT=ocr
  Default: OCR fonts are installed.

All command line switches must be given in UPPER case as designated. Example:

```
halcon-17.12.0.0-windows-runtime.exe /S /DEST="C:\HALCON"
```

2.2.2 Installing HALCON under Linux

To install HALCON on Linux systems, extract the downloaded archive in an empty directory.

```
tar zxvf halcon-17.12.0.0-linux.tar.gz
```

To start the installation, execute the shell script install-linux.sh, which is located in the top-level directory of the extracted archive, e.g., by calling

```
sh install-linux.sh
```

Step 1: Internal checks, architecture selection, license agreement

First, the setup program checks whether your system meets the requirements for running HALCON (see section 1.3 on page 8). Then, a license agreement is displayed; its text is stored in the file eula.txt, which will be installed in the directory you select for the installation.

Step 2: Select installation destination

The script will ask you in which directory you want to install HALCON; the environment variable HALCONROOT must be set to this directory later; see below. If the directory does not exist yet, the script offers to create it. On the other hand, if HALCON was already installed in the selected directory, the script asks whether to remove the old installation completely or replace existing files with the new ones.

Step 3: Select installation type

Then, you are asked to select between different installation types, which correspond to those described in section 2.2.1 on page 12. Note that you can install additional parts of HALCON at any time by extracting the archive again and copying the corresponding directories and files to the directory where you installed HALCON (see section 2.3 on page 17). You can also execute the shell script `install-linux.sh` once again to install additional parts of HALCON.

Step 4: The actual installation

Now, the actual installation starts, i.e., the requested parts are copied to the destination directory.

Step 5: Documentation language

By default, the documentation will be installed in English. The HALCON operator reference is also available in German and Japanese. You can select the desired language of this manual.

Step 6: Support for the Armv7-A platform

If you intend to cross-compile HALCON applications for Arm®-based platforms, you should also install the required libraries for the armv7a architecture.

After the installation has finished, you should perform a runtime installation on the Arm platform: Copy the file `misc/linux/halcon-17.12.0.0-armv7a-linux-runtime.tar.gz` to an empty directory on the Arm platform, and follow the first steps of this section again (extract the achive and run the installation script). See HALCON for Arm-based platforms for platform-specific information.

Step 7: Set environment variables

```
# Sample shell script for HALCON environment settings
# (sh syntax)
# If you are using the Bourne shell source this file with the following
# command:
#  source .profile_halcon

HALCONARCH=x64-linux; export HALCONARCH
HALCONROOT="/opt/halcon"; export HALCONROOT
HALCONEXAMPLES=${HALCONROOT}/examples; export HALCONEXAMPLES
HALCONIMAGES=${HALCONROOT}/examples/images; export HALCONIMAGES
PATH=${HALCONROOT}/bin/${HALCONARCH}:${PATH}; export PATH

if [ ${LD_LIBRARY_PATH} ] ; then
  LD_LIBRARY_PATH=${HALCONROOT}/lib/${HALCONARCH}:${LD_LIBRARY_PATH}; export LD_LIBRARY_PATH
else
  LD_LIBRARY_PATH=${HALCONROOT}/lib/${HALCONARCH}; export LD_LIBRARY_PATH
fi
```

Figure 2.1: Example for a shell script with environment variables in `sh` syntax, generated when installing HALCON into the directory `/opt/halcon` on a Linux system.

After the installation, the following environment variables must be set or modified in order for HALCON to work (see section A.2 on page 42 for more information about these and other environment variables):

- `HALCONROOT`: directory you installed HALCON in.
- `HALCONEXAMPLES`: directory the example programs are installed in (`$HALCONROOT/examples`).
- `HALCONIMAGES`: directory the example images are installed in (`$HALCONEXAMPLES/images`).
- `HALCONARCH`: select value corresponding to the used platform (see table 1.1 on page 8).
- `PATH`: this system variable should include `$HALCONROOT/bin/$HALCONARCH`.
- `LD_LIBRARY_PATH`: this system variable should include `$HALCONROOT/lib/$HALCONARCH`.

It is recommended to set the environment variables in a login script or a shell resource script, e.g., `.cshrc` or `.profile`. **The installation script automatically creates two example shell scripts `.cshrc_halcon` and `.profile_halcon`** in `$HALCONROOT` which contain the necessary settings in `csh` and `sh` syntax, see figure 2.1 on page 15. The suitable shell script can be included in your login script. Note that different shells offer different commands to set environment variables, e.g., `setenv <variable> <value>` or `export <variable>=<value>`. Please consult your shell's documentation for further information. If a value consists of multiple items, those items must be separated by *colons*.

Make sure `LD_LIBRARY_PATH` is set correctly after a reboot, e.g., with:

```
echo $LD_LIBRARY_PATH
```

Some systems disallow setting `LD_LIBRARY_PATH` in `.profile`. If the variable is not set after reboot, you should try to set `LD_LIBRARY_PATH` in other initialization files like `.bashrc`. Consult the documentation that comes with your operating system.

Step 8: Further configuration

Optionally, you can optimize **HALCON's automatic operator parallelization** to your computer as described in the Programmer's Guide, section 2.1.1 on page 15.

In case that you have already obtained a license from your local distributor, you can now proceed with the installation of the license file `license.dat` which has to be placed in the directory `$HALCONROOT/license`.

If the license is bound to a USB dongle, you need to allow access to the `hidraw` device for CodeMeter dongles. This can be done by copying the file `$HALCONROOT/misc/linux/udev/rules.d/59-halcon-codemeter.rules` to the `/etc/udev/rules.d` system directory (see also section 5.3.1 on page 34).

Step 9: Additional steps for Deep Learning

In order to use the Deep Learning functionality, two third-party libraries need to be installed manually (see chapter 3 on page 25).

Furthermore, the example programs for Deep Learning rely on the installation of a set of images. These images are available in a separate archive `halcon-17.12.0.0-linux-images-deep-learning.tar.gz`, which can be downloaded from `http://www.halcon.com/download`. This archive must be extracted in `$HALCONROOT`.

2.2.3 Installing HALCON under macOS

To install HALCON on macOS systems, simply double-click the downloaded installer.

Step 1: License agreement

First, a license agreement is displayed that has to be accepted to continue the installation. This license can be printed or saved to disk for later reference. After the installation has finished, the file can be found in `/Library/Application Support/HALCON-17.12/eula.txt`.

Step 2: Select installation destination

Select “Install for all users of this computer” and continue.

Step 3: Select installation type

By default, HALCON will be installed completely, including the English documentation and the supplied example programs. The HALCON operator reference is also available in German and Japanese. To select another language, or skip other parts of the installation, click “Customize”.

Step 4: The actual installation

You will need root privileges to start the installation. The installer will then copy all required files to your computer (see section A.1 on page 37 for an overview of HALCON's fileset).

Step 5: License installation (optional)

After the setup of HALCON has finished, the installer lets you install a license file. This step can be skipped if you have not yet obtained a license from your local distributor. To install the license at a later time, simply copy the obtained license file `license.dat` (or similar) to the directory `/Library/Application Support/HALCON-17.12/license`.

Step 6: Further configuration

You can optimize **HALCON's automatic operator parallelization** to your computer as described in the Programmer's Guide, section 2.1.1 on page 15. !

Apart from that, no further action on your part is required.

2.3 Installing Additional Parts of HALCON

Windows systems

On Windows systems, you can install additional parts of HALCON at any time by starting the downloaded installer again.

Linux systems

On Linux systems, you can install additional parts of HALCON at any time by starting the installer `install-linux.sh` from the downloaded archive another time.

macOS systems

On macOS systems, you can install additional parts of HALCON at any time by starting the downloaded installer again.

2.4 Manually Installing a Runtime Version of HALCON on Windows

Usually, when a machine vision application is finished it has to be set up at a production site. This involves installing HALCON runtime versions on a number of computers. Depending on that number, individual installations can become a nuisance. Quite often, a software distribution tool or script-based solution is used to install required programs on the client computers. This section shows the relevant steps of installing a HALCON runtime version manually. With this knowledge, the task of software distribution can be automated.

As an alternative, a **separate runtime installer is available for download**. This stripped-down installer features a silent installation mode (see section 2.2.1.2 on page 14). See section 2.1 on page 11 for download information. ☜

You need administrator privileges to perform the manual installation.

1. Set the environment variables `HALCONROOT` and `HALCONARCH`. See also section A.2 on page 42.

2. Copy the required runtime DLLs to, e.g., `C:\%HALCONROOT%\bin\%HALCONARCH%` (see table 1.1 on page 8 for values of the environment variable `HALCONARCH`). The following DLLs are required: `halcon.dll`, the relevant HALCON language interface used by the application, e.g., `halconcpp.dll`, if used, the appropriate image acquisition device interface, and the required I/O device interface.

   If you want to use HALCON/.NET, you must also copy the corresponding assemblies to the directory `%HALCONROOT%\bin\dotnet20`, or `%HALCONROOT%\bin\dotnet35` (also see the Programmer's Guide, section 10.1 on page 59).

   If the application uses HALCON XL, the relevant DLLs or assemblies with the suffix `xl` must be copied instead.

   The HALCON DLLs must never be found in the `%PATH%` twice or even multiple times. Additionally, it is highly recommended not to copy the HALCON DLLs into the Windows system directories (`C:\%WINDIR%`, `C:\%WINDIR%\system`, and `C:\%WINDIR%\system32`).

3. Add the directory with the HALCON DLLs to the `%PATH%` environment variable, e.g., `C:\%HALCONROOT%\bin\%HALCONARCH%`. This step is not required if the application resides in the same directory as the DLLs.

4. Copy the HALCON help files to `%HALCONROOT%\help`.

5. Copy the license file `license.dat` (or similar, see section 1.4 on page 10) to `%HALCONROOT%\license`.

6. Depending on the application, one or more of the following directories need to be copied to `%HALCONROOT%` as well: `calib`, `filter`, `lut`, `ocr`, `procedures`.

7. If you intend to use the GenICamTL or USB3 Vision interface, copy the directory `genicam` to `%HALCONROOT%` as well.

8. If you intend to use the GigEVision interface, copy the directory `genicam` to `%HALCONROOT%` and install the MVTec GigE Vision Streaming Filter using the corresponding executable (`GevStreamingFilter.exe` or similar) from the directory `misc`. Use the command line option `--help` for further instructions and return values.

2.5 Updating HALCON

With HALCON, the term “update” means to install a newer maintenance release over a release based on the same HALCON version. As described in section 1.2 on page 7, you can update HALCON without needing a new license.

When updating an existing HALCON installation under Windows, the downloaded installer will ask you whether you want to replace the already installed components with the updated ones.

2.6 Upgrading HALCON

With HALCON, the term “upgrade” means to install a newer major version over an older one. As described in section 1.2 on page 7, new versions are not compatible to older ones. Therefore, we recommend to **uninstall the older HALCON release before installing the new one**. On Windows systems, this is especially important in order to clean up the registry.

The easiest way to uninstall the older version on Windows is to start the installation of the new version, because the setup program offers to perform the uninstallation for you if it detects an older version. More information regarding uninstalling HALCON can be found in section 2.8 on page 19. If you want to keep the older version, please refer to section 2.9 on page 21.

The installation process itself is the same as the first-time installation described in section 2.2 on page 11. Please note that after installing the new version you must upgrade the license as described in section 4.5 on page 31.

2.7 HALCON Installer Switches under Windows

The installation process under Windows can be customized by calling the setup program with designated switches. This can be done from a Windows command prompt, or from a custom batch file. The following switches are recognized:

`/CPU`=[x86sse2 | x64]	Force architecture.
`/LANG`=[english | german | japanese]	Force language selection.

Also, a silent runtime installer exists which provides additional options for automated installations (see section 2.2.1.2 on page 14). It can be downloaded from `http://www.halcon.com/download`.

2.8 Uninstalling HALCON

2.8.1 Uninstalling HALCON under Windows

The preferred method to uninstall HALCON is to use the automatic uninstallation program as described in the following section. In case you want to keep track of what is happening to your system, you may want to follow the instructions given in section 2.8.1.2 on page 19.

2.8.1.1 Uninstalling Automatically

There are multiple ways to uninstall HALCON; please note that not all are available for older versions of HALCON:

1. Select `Start ▷ Programs ▷ MVTec HALCON 17.12 ▷ Uninstall HALCON` in the Windows start menu.
2. Choose `Add/Remove Programs` in the system control panel.
3. When installing HALCON, the setup program checks whether there is an old version of HALCON and allows you to uninstall it. Note that during updating an existing HALCON installation under Windows the setup program will ask you whether you want to replace the already installed components with the updated ones.

Typically, this is all you have to do. If you encounter any problems, please refer to section 5.2 on page 33.

Note that **the uninstallation removes exactly those files that were installed**. This has two implications: If you added files after the installation manually, e.g., new image acquisition interfaces, extension packages, images, or manuals, these files and the corresponding directories will “survive” the uninstallation. On the other hand, if you only modified a file, e.g., an example, without changing its name the uninstallation will remove it nevertheless. Therefore you might want to copy such files to another directory before starting the uninstallation.

The uninstallation process will not remove any *user-specific settings*. This means that entries concerning, e.g., the layout of HDevelop or its file history, will be left in the file `%APPDATA%\MVTec\HDevelop.ini`. If you have run the utility hcheck_parallel, AOP information has been stored in `%ProgramData%\MVTec\HALCON 17.12\.aop_info`. You may remove these files manually without risk. Moreover, the uninstaller does not remove the MVTec GigE Vision streaming filter driver if it is installed. This driver needs to be removed separately.

2.8.1.2 Uninstalling Manually

The commands given in the following description should be entered in a Windows command prompt, which can be obtained by entering `cmd.exe` in the dialog `Start ▷ Run`. You need administrator privileges to perform the uninstallation.

1. For extensive cleaning of the Windows registry you can optionally run the program `%HALCONROOT%\misc\x86-win32\clean_reg_halconx.exe`.
2. Delete the installation directory. You can also use Windows Explorer to do this. Please note that **the license file and any local additions to this directory will be lost**. A backup of these files is highly recommended.

   ```
   rmdir /S "%HALCONROOT%"
   ```

3. Delete all HALCON registry keys. Replace `x.x` at the end of the following command with the version number you are uninstalling. Alternatively, start `regedit.exe` to use the Windows registry editor to delete the keys.

   ```
   reg delete HKLM\SOFTWARE\MVTec\HALCON\x.x}             (Windows)
   reg delete HKLM\SOFTWARE\Wow6432Node\MVTec\HALCON\x.x} (Windows x64)
   ```

   You can query all installed versions of HALCON using this command:

   ```
   reg query HKLM\SOFTWARE\MVTec\HALCON                   (Windows)
   reg query HKLM\SOFTWARE\Wow6432Node\MVTec\HALCON       (Windows x64)
   ```

4. Delete all environment variables set by HALCON. The indented lines must be appended to the preceding lines separated by a space character. See section A.2 on page 42 on how to edit environment variables using the Windows GUI. Please also use the GUI to manually remove the HALCON binary directory from the environment variable PATH.

```
reg delete "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment"
           /V HALCONROOT
reg delete "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment"
           /V HALCONARCH
reg delete "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment"
           /V HALCONEXAMPLES
reg delete "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment"
           /V HALCONIMAGES
```

5. Delete the uninstall information created when HALCON was installed. Replace <ID> with the actual HALCON GUID, which depends on the installed version:
HALCON 6.1 → BBEC9F40-4A36-11D6-A14C-00E0296C2846
HALCON 7.0 → 47F424B4-1077-11D8-A0D3-00E01883F42C
HALCON 7.1 → 0B2DE0B7-FD31-11D9-A19F-00E01883F42C
HALCON 8.0 → 096CE3F7-2FD9-4460-A270-F9F4740DB91B

```
rmdir /S "%ProgramFiles%\InstallShield Installation Information\{<ID>}"
reg delete HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{<ID>}
```

For HALCON 9.0 or higher, please use the following key:

```
reg delete "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\
MVTec HALCON xxx"                                      (Windows)
reg delete "HKLM\SOFTWARE\Wow6432Node\\Microsoft\Windows\CurrentVersion\
Uninstall\MVTec HALCON xxx"                            (Windows x64)
```

where xxx has to be replaced with the actual version number, e.g., 12.0.

2.8.2 Uninstalling HALCON under Linux

HALCON provides no uninstallation script for Linux systems, therefore you must perform the uninstallation manually.

Please note: The following procedure will **delete your local additions to the HALCON base directory**. To check for any local additions and changes beforehand, extract the downloaded archive, e.g., into /tmp/halcon and run

```
diff -q -r /tmp/halcon $HALCONROOT | grep $HALCONROOT
```

The actual uninstallation consists of simply removing the content of the HALCON base directory $HALCONROOT and all its subdirectories, e.g., by executing

```
rm -rf $HALCONROOT
```

Furthermore, remove the subdirectory .hdevelop of the directory referenced by the environment variable HOME (see section A.2 on page 42); HDevelop creates this directory to save options, window positions, and the file history. If you have run the utility hcheck_parallel, AOP information has been stored in $HALCONROOT/.aop_info. This file can also safely be deleted.

Finally, delete references to HALCON from the environment variables (see section 2.2.2 on page 14).

2.8.3 Uninstalling HALCON under macOS

To uninstall HALCON from macOS systems, run the provided shell script. The 'x.x' in the middle of the following path should represent the version number you want to uninstall (e.g. 12.0). The same applies to 'halconx', where the x should be replaced by the version number (e.g. halcon12).

```
/Library/Application\ Support/HALCON-x.x/bin/remove_halconx_installation.sh
```

For example, if you want to uninstall HALCON 12, you have to run the following shell script:

```
/Library/Application\ Support/HALCON-12.0/bin/remove_halcon12_installation.sh
```

Note that the script for uninstalling HALCON 11.0 is for the first time provided with HALCON 12.0.

This script will simply remove the content of the various HALCON directories and their subdirectories and discard the corresponding package receipt data.

The script will not remove the preference file that HDevelop creates to save options, window positions and the file history. To remove this file, run the following command in Terminal.app:

```
rm $HOME/Library/Preferences/com.mvtec.HDevelop.plist
```

2.9 Managing Multiple HALCON Versions

Linux systems

On Linux systems, you can switch between different HALCON versions by setting the environment variable HALCONROOT accordingly. Note that in order for this method to work, paths based on HALCONROOT in other environment variables like PATH and LD_LIBRARY_PATH must use the variable and not its content. See figure 2.1 on page 15 for an example.

Windows systems

Under Windows, you must adapt those environment variables that are set during the installation, i.e., HALCONROOT, HALCONARCH, PATH, HALCONEXAMPLES, and HALCONIMAGES, and those you set yourself (e.g., HALCONEXTENSIONS). Please refer to section A.2 on page 42 for more information about setting environment variables.

2.10 Switching between Different HALCON Platform Versions under Windows

As described in section 1.3 on page 8, HALCON is provided for different platforms. However, only the first installation sets the environment variable PATH (see section A.2 on page 42). If you want to switch e.g., from Windows 32-bit to Windows 64-bit, you must adapt the environment variable HALCONARCH accordingly (see table 1.1 on page 8), which is referenced in PATH.

2.11 Installing HALCON I/O Device and Image Acquisition Interfaces

With every HALCON installation, you obtain several already installed I/O device and image acquisition interfaces (see also section A.1 on page 37). Additional interfaces are obtainable from the MVTec/HALCON Download area.

In between HALCON releases, image acquisition interfaces might be updated by MVTec or the manufacturer of an image acquisition device. Such updates are indicated on MVTec's WWW server, to which you can connect by selecting HDevelop's menu entry Help ▷ HALCON News (WWW)or in the Start Dialog which appears when starting HDevelop. You can then download the interface together with its documentation and HDevelop example programs.

2.12 Installing HALCON Extension Packages

The HALCON Extension Package Interface enables you to integrate newly developed image processing algorithms into HALCON in the form of so-called *extension packages*. The same mechanism is used by MVTec to extend the current HALCON release with additional functionality. Which extensions packages are currently available can be checked by selecting HDevelop's menu entry `Help` ▷ `HALCON News (WWW)`, which connects to MVTec's WWW server.

This section describes how to integrate a (downloaded) package named `newextpkg` in order to use it within your HALCON system.

First, unpack the package to a directory of your choice, e.g., `%HALCONROOT%`. Then, add the *complete* path of the package, e.g.,

```
%HALCONROOT%\packages\newextpkg
```

to the environment variable `HALCONEXTENSIONS`. Note, that the delimiter between paths in an environment variable is a semicolon on Windows systems and a colon on Linux/macOS systems.

Never change the name of an extension package or the corresponding names of the libraries or DLLs contained in it. These names are encoded *within* the libraries/DLLs. If you change the names this information will no longer match. Thus, the loader of the operating system will fail to open the dynamic libraries.

If the package contains images used, e.g., within example programs we recommend to include the (complete) path to the corresponding directory `images` within the package in the environment variable `HALCONIMAGES` (see section A.2 on page 42) to access those images without specifying a complete path.

2.12.1 Using an Extension Package Within HDevelop

In order to use a new package within HDevelop under Windows/macOS, you just need to restart the program. HDevelop automatically integrates all extension packages specified in `HALCONEXTENSIONS`, i.e., the operators contained in a package can be accessed and used like any other HALCON operator.

Under Linux, you must include the package library subdirectory (i.e., `lib/$HALCONARCH`) in the environment variable `LD_LIBRARY_PATH` before starting HDevelop the first time (see table 1.1 on page 8 for the possible values of `HALCONARCH`).

2.12.2 Using an Extension Package in a Stand-Alone Application

If you want to generate a stand-alone application that uses an extension package, you have to link the package libraries (DLLs under Windows, shared libraries under Linux/macOS) to the application code, in addition to the HALCON library.

2.12.2.1 Using an Extension Package Under Windows

In order to create new application programs you have to link the corresponding language interface library, e.g., `packagecpp.lib` for a C++ application, to your objects. Furthermore, you will need the HALCON interface library, in the example of a C++ application `halconcpp.lib`, as for any HALCON application.

To be able to link the package DLL to your application program, the *complete* DLL file path of the new package, e.g.,

```
%HALCONROOT%\packages\newextpkg\bin\%HALCONARCH%
```

must be added to the environment variable `PATH` (see table 1.1 on page 8 for the possible values of `HALCONARCH`).

Do not copy a package DLL into the Windows system directories, as it would be loaded twice in this case!

2.12.2.2 Using an Extension Package Under Linux

In order to create new application programs, you must link libnewextpkg.so and the corresponding language interface library, e.g., libnewextpkgcpp.so for a C++ application, to your objects (besides libhalcon.so and the HALCON interface library, in the example of a C++ application libhalconcpp.so, as for any HALCON application).

Furthermore, you have to add the path to the package library subdirectory lib/$HALCONARCH to the environment variable LD_LIBRARY_PATH, otherwise the loader will fail to access the libraries.

2.12.2.3 Using an Extension Package Under macOS

In order to create new application programs, you must link libnewextpkg.dylib and the corresponding language interface library, e.g., libnewextpkgcpp.dylib for a C++ application, to your objects (besides libhalcon.dylib and the HALCON interface library, in the example of a C++ application libhalconcpp.dylib, as for any HALCON application).

Chapter 3

Installing Third-Party Components

This chapter describes how to install third-party components that might be required in order to use a specific functionality of HALCON. Currently, this is limited to users of **Deep Learning** with Convolutional Neural Networks. Please note that either of the Deep Learning modules Inference or Training must be licensed.

In addition to a valid license, the following third-party components are required:

- NVIDIA GPU and up-to-date graphics driver (see section 1.3 on page 8 for the system requirements),
- the NVIDIA CUDA Basic Linear Algebra Subroutine library from the CUDA Toolkit (cuBLAS),
- the NVIDIA CUDA Deep Neural Network library (cuDNN).

3.1 Getting the Components

Deep Learning is currently supported on Windows x64 and Linux x86_64.

- Download and install the CUDA Toolkit from `https://developer.nvidia.com/cuda-toolkit`. The CUDA Toolkit contains the cuBLAS library. Currently, the cuBLAS libraries included in CUDA 8 and 9 are supported.
- Download the cuDNN library from `https://developer.nvidia.com/cudnn`.
  You need to register for the NVIDIA Developer Program first. Currently, only cuDNN 7 is supported. Extract the downloaded archive to a temporary folder.

3.2 Copying the Libraries

After installing/unpacking the components, the libraries need to be copied to a location where HALCON can find them. First, you need to locate the DLLs or shared objects for Windows or Linux, respectively.

The cuBLAS library (`cublas64_90.dll` or `libcublas.so`) can be found where the CUDA Toolkit has been installed.

The cuDNN library (`cudnn64_7.dll` or `libcudnn.so`) is contained in the extracted archive.

In general, the DLL for Windows is contained in the `bin` subdirectory. The shared object for Linux is contained in the `lib64` subdirectory.

On Windows: Copy the two library files to `%HALCONROOT%\bin\%HALCONARCH%\thirdparty`.

On Linux: Copy the two library files to `$HALCONROOT/lib/$HALCONARCH/thirdparty`.

If the subdirectory `thirdparty` does not exist, create it first.

Alternatively, it is possible to adapt the environment variables (`LD_LIBRARY_PATH` for Linux, `PATH` for Windows) to include the respective libraries.

Chapter 4

All About HALCON Licenses

Section 1.4 on page 10 already contained an overview of the possible licensing schemes. In this chapter, you will find detailed information about how to obtain and install

- evaluation licenses (section 4.2 on page 28),
- development licenses (section 4.3 on page 29), and
- runtime licenses (section 4.4 on page 30),

Finally, section 4.5 on page 31 shows how to upgrade a license.

4.1 What is a License?

HALCON licenses are stored in so-called *license files*. The content of these files specifies

- what is licensed (e.g., development version, runtime version, etc.)
- which modules are licensed (e.g., Calibration, Matching, etc.)
- whether the license is temporary (e.g., evaluation license) or permanent
- the hardware to which the license is bound (see below)

License files are named `license.dat` (or `license-17.12.dat` or similar, see section 1.4 on page 10) and reside in the subdirectory `license` of the folder where you installed HALCON. **Note that HALCON will not run if you modify the license keys within the license file manually!**

!

A single license allows to run a specific version of HALCON by one user on one system.

The system is identified by a unique Host-ID. Depending on the type of Host-ID you may install HALCON on any number of computers, but in any case you must use it only on the one computer to which the Host-ID is attached. Further information about licenses can be found in the EULA.

The license is not bound to a specific user. Consequently, different users may use the licensed HALCON version, but not simultaneously. To have multiple users run HALCON at the same time, the corresponding number of licenses are needed. Note that each system service or daemon which executes HALCON is also counted as user. Thus, if only one license is available and a system service executes HALCON, the regular user is not able to run another instance of HALCON in parallel.

Network Card Versus Dongle Binding

As noted in section 1.4 on page 10, development and runtime licenses are bound to a certain hardware component. This is either the *network card* (see section 4.3.1 on page 29) or a *dongle* (see section 4.3.2 on page 29).

Dongle-bound licenses allow to use HALCON on different computers by moving the dongle. Of course, network cards can also be switched between computers, but in practice they can be regarded as fixed. Thus, if you want to develop applications with HALCON on more than one stand-alone computer at different times, the easiest solution is to obtain a dongle-bound license.

Identifying the Hardware

The license manager software identifies a network card or a HALCON dongle by a so-called *host ID*. A valid host ID is the unique, immutable, machine-readable identification of an actual piece of hardware as devised by the hardware vendor.

HALCON includes a tool to query the host IDs from the command line. Open a Windows Command Prompt or a Linux/macOS shell, and enter the following command:

```
hhostid
```

This command produces a list of host IDs, which are available for licensing. For example:

```
0d00cafebabe 3-3123456
```

Use `hhostid -i` to get additional information, such as the type of a certain host ID:

```
hhostid -i
#0: 0d00cafebabe [MAC] <Local Area Connection>
#1: 3-3123456 [CM] <CM Container>
```

Each output line lists the following properties: First, the running number followed by the actual host ID is displayed. In the case of network adapters, the *permanent* MAC address is displayed. The host ID type is displayed in square brackets (`MAC` for network adapters, `CM` for HALCON dongles). The value in angle brackets shows the interface name as reported by the operating system. If the currently assigned MAC address of a network adapter differs from its permanent address, it would be reported here as well. HALCON only supports MAC based licenses bound to universally administered, permanent MAC addresses. Locally administered MAC addresses are rejected.

See section 5.3.1 on page 34 for a detailed description on solving problems extracting the host ID.

HDevelop automatically checks whether any network cards or HALCON dongles are present and displays their host IDs in the menu item:

```
Help ▷ About.
```

For example, the following information will be displayed on a computer equipped with a network card and a dongle:

```
HALCON 17.12 Progress

HDevelop version: 17.12 (dd.mm.2017)
HALCON version: 17.12 (dd.mm.2017)

Platform version: x64-win64 (avx2)

(c) 1996-2017
MVTec Software GmbH, Munich, Germany

The host IDs of this computer are:
0d00cafebabe                    (network card ID)
3-3123456                       (dongle ID)
```

4.2 Evaluation Licenses

As already noted in section 1.4 on page 10, with an evaluation license you can evaluate the full functionality of HALCON free of charge on any computer. The only restrictions are that evaluation licenses are valid only for a limited time (typically a month), and no commercial applications may be developed.

Step 1: Obtain the license

You can obtain an evaluation license from your local distributor.

Step 2: Install the license

“Installing” the license simply means placing the license file into the subdirectory `license` of the folder where you installed HALCON. If necessary, rename the file to `license.dat` (or `license-17.12.dat` or similar, see section 1.4 on page 10).

Note that you can evaluate HALCON on any computer where you installed HALCON just by copying the evaluation license file into the corresponding subdirectory `license`. You can also evaluate HALCON under different operating systems.

4.3 Development Licenses

Like the evaluation license, a development license allows you to use the full functionality of HALCON including the development tools like HDevelop (see also section 1.4 on page 10). However, it must be bound to a certain hardware component (see also section 4.1 on page 27). The following sections describe how to proceed to obtain and install a

- license bound to a network card (section 4.3.1)
- license bound to a dongle (section 4.3.2 on page 29)

4.3.1 License Bound to a Network Card

Step 1: Extract the host ID

The easiest way to extract the host ID is to execute the following command from a Windows Command Prompt or a Linux/macOS shell.

```
hhostid -i
```

All host IDs of type MAC can be used for licensing. See section 4.1 on page 28 for more information.

Step 2: Obtain the license

Send the host ID of the network card to your local distributor. The distributor then sends you a *license file*.

Step 3: Install the license

Place the license file into the subdirectory `license` of the folder where you installed HALCON. If necessary, rename the file to `license.dat` (or `license-17.12.dat` or similar, see section 1.4 on page 10).

4.3.2 License Bound to a Dongle

Step 1: Obtain dongle and license

Please note that **you cannot use any dongle but only those supplied by MVTec via your local distributor**. Currently, HALCON supports USB dongles.

The distributor will send you the dongle together with a corresponding *license file*. The dongle ID is printed on the dongle.

Step 2: Install the license

Place the license file into the subdirectory `license` of the folder where you installed HALCON. If necessary, rename the file to `license.dat` (or `license-17.12.dat` or similar, see section 1.4 on page 10).

If you want to use HALCON on more than one computer by switching the dongle between them, repeat this step for every computer.

4.4 Runtime Licenses

In contrast to a development license, a runtime license only allows to run HALCON applications. Furthermore, a runtime license can be restricted to a set of selected modules if your application does not require the full functionality of HALCON. The list of required modules of an application is determined in the following way:

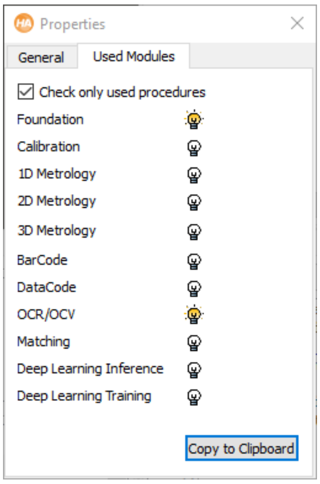
Step 1: Extract the required modules

To extract the modules that are used by an application proceed as follows:

1. If the application is running in HDevelop, select the menu item `File` ▷ `Properties`, which will open a dialog. In its tab `Used Modules` the used modules are listed (see the HDevelop User's Guide, section 6.2.1.16 on page 55, for more information). Figure 4.1 shows the result for an OCR application.

   If you click `Copy to Clipboard`, the required modules are saved in the clipboard, from where you can insert them in other applications.

   Please note that this method determines the list of used modules by inspecting *all* operators of the current program, no matter if they can be reached or not. If the program contains operator calls that are never executed, it is recommended to deactivate the corresponding program lines using F4 before opening this dialog to get a correct list of used modules.



☑ Check only used procedures

Foundation	●
Calibration	○
1D Metrology	○
2D Metrology	○
3D Metrology	○
BarCode	○
DataCode	○
OCR/OCV	●
Matching	○
Deep Learning Inference	○
Deep Learning Training	○

Figure 4.1: Used modules for an OCR application.

2. If the application is written in a programming language (C, C++, C#, VB.NET, etc.), insert the operator `get_modules` (see the corresponding entry in the HALCON Reference Manuals for more information) at the end of the program. Note that **the operator `get_modules` will only return the correct modules if all HALCON operators used in the application are executed at least once**.

Step 2: Extract the host ID

Please refer to section 4.3.1 on page 29 (network card) for information about how to extract the host ID. If you choose a dongle-bound license, no further action is required as you get the dongle together with the license (see section 4.3.2 on page 29).

Step 3: Obtain the license

Send the determined module names and – except in case of a dongle-bound license – the extracted host ID to your local distributor.

The distributor then sends you a license file. If you requested a dongle-bound license, you will also receive the dongle.

Step 4: Install the license

Place the license file into the subdirectory `license` of the folder where you installed HALCON. If necessary, rename the file to `license.dat` (or `license-17.12.dat` or similar, see section 1.4 on page 10).

4.5 How to Upgrade a License

If you upgrade a HALCON license to a newer version, e.g., from HALCON 12.0.x to HALCON 13.0, your distributor provides you with a new license file which contains new license keys. This new license file should replace the old one in `%HALCONROOT%\license\license.dat`.

4.6 How to Check the Genuineness of HALCON (Windows)

To verify that your HALCON installation is genuine, you can check the digital signature of the file `halcon.dll`. Make sure that your Windows installation is fully updated. Otherwise, the signature verification might not work as expected.

- Right-click on the file `%HALCONROOT%\bin\%HALCONARCH%/halcon.dll`.
- Select “Properties”.
- Select the tab “Digital Signature”.
- Select the signature “MVTec Software GmbH” and click on “Details”.

If your version of HALCON is genuine, the dialog reads “This digital signature is OK”. Otherwise, your version of HALCON has been tampered with. This might lead to unexpected behavior of HALCON.

Chapter 5

Troubleshooting

This chapter offers help for problems encountered during installing or uninstalling HALCON on Windows systems (section 5.1 on page 33 and section 5.2 on page 33, respectively), with the licensing mechanism (section 5.3 on page 34), or when starting HDevelop or your own HALCON applications (section 5.4 on page 35) and other miscellaneous problems.

Note that throughout the chapter the environment variable HALCONARCH is referenced. See table 1.1 on page 8 for the possible values of this variable.

5.1 Problems During Installation (Windows)

- **Installer fails to detect SSE2**
  In rare cases the installer fails to detect the SSE2 capabilities of the processor. You can force the installation of the SSE2 version by calling the installer with the command line option /CPU=x86sse2. For example:

  ```
  E:\install-windows\setup.exe /CPU=x86sse2
  ```

  See section 2.7 on page 18 for information about installer switches.

5.2 Problems During Uninstallation (Windows)

- **Uninstallation failed**
  If the automatic uninstallation fails for another reason, proceed as follows:
  - Start regedit and delete the following key:

    On 32-bit Windows:

    ```
    HKEY_LOCAL_MACHINE -> SOFTWARE -> MVTec -> HALCON -> x.x
    ```

    On 64-bit Windows:

    ```
    HKEY_LOCAL_MACHINE -> SOFTWARE -> Wow6432Node -> MVTec -> HALCON -> x.x
    ```

  - **For recent versions (HALCON 9.0 or higher)**
    Also delete the following key:

    On 32-bit Windows:

    ```
    HKEY_LOCAL_MACHINE -> SOFTWARE -> Microsoft -> Windows ->
      CurrentVersion -> Uninstall -> MVTec HALCON x.x
    ```

    On 64-bit Windows:

    ```
    HKEY_LOCAL_MACHINE -> SOFTWARE -> Wow6432Node -> Microsoft -> Windows ->
      CurrentVersion -> Uninstall -> MVTec HALCON x.x
    ```

- **Only for HALCON 8.0**
  Also delete the key

  ```
  HKEY_LOCAL_MACHINE -> SOFTWARE -> Microsoft -> Windows ->
    CurrentVersion -> Uninstall ->
   {096CE3F7-2FD9-4460-A270-F9F4740DB91B}
  ```

- **Only for HALCON 7.1**
  Also delete the key

  ```
  HKEY_LOCAL_MACHINE -> SOFTWARE -> Microsoft -> Windows ->
    CurrentVersion -> Uninstall ->
   {0B2DE0B7-FD31-11D9-A19F-00E01883F42C}
  ```

- **Only for HALCON 7.0**
  Also delete the key

  ```
  HKEY_LOCAL_MACHINE -> SOFTWARE -> Microsoft -> Windows ->
    CurrentVersion -> Uninstall ->
    {47F424B4-1077-11D8-A0D3-00E01883F42C}
  ```

- **Only for HALCON 6.1**
  Also delete the key

  ```
  HKEY_LOCAL_MACHINE -> SOFTWARE -> Microsoft -> Windows ->
    CurrentVersion -> Uninstall ->
    {BBEC9F40-4A36-11D6-A14C-00E0296C2846}
  ```

- **Only for HALCON releases < 6.1**
  Also delete the key

  ```
  HKEY_LOCAL_MACHINE -> SOFTWARE -> Microsoft -> Windows ->
    CurrentVersion -> Uninstall -> HALCON
  ```

- Using, e.g., the Windows Explorer, delete the directory

  ```
  %ALLUSERSPROFILE%\Start Menu\Programs\MVTec HALCON x.x
  ```

- Delete the environment variables `HALCONROOT` and `HALCONIMAGES` in `Start` ▷ `Settings` ▷ `Control Panel` ▷ `System` (Windows Vista and Windows 7: ... ▷ `System` ▷ `Advanced System Settings`).
- Finally, delete the contents of the HALCON directory and all its subdirectories.

5.3 Problems Concerning Licenses

If you encounter problems with your HALCON license even though your license file exists and is located in the correct directory, a first step is always to check if the host ID identifying your network card or dongle matches the entry in the license file (see the corresponding sections in chapter 4 on page 27). If the two do not match, please send the new identifying information to your distributor. See section 5.3.1 if you encounter problems with extracting the identifying information.

5.3.1 Extracting Host IDs

- **`hhostid` does not return a dongle ID**
  Please note that only HALCON dongles (orange) are reported by `hhostid`. Other dongles (e.g., blue MERLIC dongles) are not supported.

- hhostid **does not return a MAC address**

  This might happen if you call hhostid from a virtualized environment using locally administered MAC addresses. HALCON only supports MAC based licenses bound to universally administered, permanent MAC addresses. Locally administered MAC addresses are rejected.

- hhostid **shows no dongle ID on Linux even though a HALCON dongle is plugged in**

  On Linux, the hidraw device must exist. This device is usally available on systems using the default kernel configuration. However, on embbeded systems, this device might not be configured, in which case hhostid would not be able to access the dongle ID. The solution is to enable the hidraw device in the kernel configuration (CONFIG_HIDRAW=y) and to build a custom kernel.

  In addition, HALCON needs read/write permissions on the hidraw device. The following udev rule is required to achieve this:

  ```
  # Udev rules to allow access to hidraw device for CodeMeter dongles

  ACTION=="remove", GOTO="halcon-cm_end"

  SUBSYSTEM=="hidraw", ATTRS{idVendor}=="064f", ATTRS{idProduct}=="2af9", MODE="666"

  LABEL="halcon-cm_end"
  ```

  You can add this rule to /etc/udev/rules.d yourself, or copy the file $HALCONROOT/misc/linux/udev/rules.d/59-halcon-codemeter.rules to that directory.

5.4 Troubleshooting in HDevelop or HALCON Applications

5.4.1 Startup Errors

This section explains miscellaneous error messages when starting HDevelop or your own HALCON applications and their reasons.

- **Error using license file**

  This error message might have several reasons:

  - The file %HALCONROOT%\license\license.dat is missing and/or not readable.
  - Your license is not valid on this machine.
  - If there is more than one user trying to use HALCON via remote access, the second user gets an error message.

- **No license for this operator**

  The operator which you try to execute belongs to a HALCON module that is not licensed (see section 4.4 on page 30. Obtain a new license including this module.

- **hdevelop: Command not found** (Linux)

  Check your system environment variable PATH. It must include the path $HALCONROOT/bin/$HALCONARCH.

- **lib* : can't open file** (Linux)

  Check the system variable LD_LIBRARY_PATH (see section A.2 on page 42).

- **No help files for package <package-name> in directory <directory>**

  Possible reasons for this error message are:

  - No files %HALCONROOT%\help\* (if the package name is "system") or no help files in one of the user packages.
  - If the package name is "system": Wrong HALCONROOT.
  - Check the file permissions. Probably HDevelop cannot access important files.

- **Help file for package <package-name> is corrupt**

  Possible reasons for this error message are:

  - If the package name is "system": Inconsistent version of %HALCONROOT%\help\* or wrong HALCONROOT.
  - If the package name is that of a user package: Inconsistent version of the help files of this package.

- **Can't open display** (Linux)

  If you see an error message like this you may have a wrong system variable DISPLAY and/or your program is not allowed to open a window by the specified X-server.

- **No refresh of window content on a Linux system**

  On some Linux systems the default behavior regarding occluded windows may be set in an inconvenient way for HALCON. The result is that if a window is temporarily occluded by another window, its content is not saved and restored anymore, i.e., windows remain "black" after uncovering. An example for this are all SuSE Linux distributions $\geq$ 7.0. The corresponding property is called "backing-store"; you can check the current setting of this property by typing (the following example corresponds to a SuSE 8.2 Linux system):

  ```
    xdpyinfo | grep backing-store
  ```

  which should result in the output like

  ```
    options:    backing-store YES, save-unders YES
  ```

  if the window content is saved and restored. You can change this behavior by changing the configuration file of your X server. It usually resides in /etc/X11/xorg.conf if you are using Xorg, or in /etc/X11/XF86Config if you are using XFree86. Consult your system's documentation if in doubt.

  You will probably need to become root to modify this file. Open the file in a text editor, find the section named "Device", and add the following option to this section:

  ```
  Section "Device"
    ...
    Option        "BackingStore" "True"
  EndSection
  ```

  Alternatively, you can modify the file Xservers residing in the directory /usr/lib/X11/xdm (or /opt/kde3/share/config/kdm in case of newer Linux versions), see your system's documentation. Note, that you probably need root privileges to modify this file. Append the option +bs (i.e., "plus backing-store") to the line that starts the local X server:

  ```
    :0 local /usr/X11R6/bin/X :0 vt07 +bs
  ```

  Now, stop and start the X server again (by using the appropriate commands or by rebooting your computer); the command xdpyinfo now should yield the output shown above.

Appendix A

More on the Installation

This appendix contains information about

- the installed file structure (section A.1),
- the relevant environment variables (section A.2 on page 42),
- the registry keys on Windows systems (section A.3 on page 44), and

A.1 The Installed File Structure

A.1.1 Main Directory

Let's take a look at the installed file structure in the directory designated by the environment variable HALCONROOT. The default location suggested by the installer is operating system-dependent:

Windows	C:\Program Files\MVTec\HALCON-17.12
Linux	/opt/halcon
macOS	/Library/Frameworks/HALCON.framework (most of the following files are installed here unless stated otherwise)

In the following, the most important directories and files are described briefly. Please note that, depending on your installation, not all directories may be present.

A.1.1.1 Windows

The default location for Windows is C:\Program Files\MVTec\HALCON-17.12:

bin

Contains HALCON programs, for example HDevelop, in subdirectories corresponding to the different platforms. This directory also contains the DLLs of the HALCON libraries, the DLLs for the supported image acquisition interfaces, and I/O device interfaces. The subdirectories dotnet20 and dotnet35 contain the HALCON/.NET assemblies based on .NET Framework 2.0, and .NET Framework 3.5, respectively.

calib

Contains description files for the calibration plates, which you can use to calibrate your camera.

doc\html\manuals

Contains some of the User's Manuals.

doc\html\reference\acquisition

Contains the HTML documentation of the HALCON Image Acquisition Interfaces.

`doc\html\reference\io`
: Contains the HTML documentation of the I/O device interfaces.

`doc\html\reference\operators`
: Contains the HALCON Operator Reference.

`doc\index`
: Contains the index files that are relevant for the full-text search of the documentation.

`doc\macros`
: Contains the style files for the HALCON Operator Reference in all translated languages. These files set the layout and provide macros for names and formatting.

`doc\pdf\manuals`
: Contains the PDF version of the User's Manuals.

`doc\pdf\reference`
: Contains the PDF version of the HALCON Operator Reference in HDevelop syntax.

`doc\pdf\solution_guide`
: Contains the PDF version of the Solution Guide I, the Solution Guide II, and Solution Guide III.

`filter`
: Contains predefined filter masks.

`genicam`
: Contains the underlying GenAPI runtime software for the GigEVision, GenICamTL, and USB3Vision image acquisition interfaces.

`help`
: The files in this directory act as the HALCON database, i.e., they provide information about HALCON to HDevelop and to all HALCON applications. In particular, they contain the operator database. The XML files starting with `index_manuals` contain the index data of the manuals. The XML files starting with `index_examples` contain the data for the `Browse Examples` dialog in HDevelop and those starting with `tip_of_the_day` contain the information for the Tip of the Day that appears in the HDevelop Start Dialog.

`include`
: Contains the header files that are necessary to use HALCON within the programming languages C or C++.

`lib`
: Contains the HALCON libraries.

`license`
: The license file must be placed here (see section 4.1 on page 27).

`lut`
: Contains predefined look-up tables.

`misc`
: Contains miscellaneous files for the installer, e.g., the GigE Vision filter driver. In addition, the installer of the HALCON `Variable Inspect` can be found here. Please refer to the Programmer's Guide, section 3.3 on page 24, for more information about this extension.

`ocr`
: Contains pretrained fonts.

`procedures`
: Contains external procedures for HDevelop and HDevEngine.

A.1.1.2 Linux

The default location for Linux is `/opt/halcon`:

`bin`

Contains HALCON programs, for example HDevelop, in subdirectories corresponding to the different platforms. The subdirectories `dotnet20` and `dotnet35` contain the HALCON/.NET assemblies based on .NET Framework 2.0, and .NET Framework 3.5, respectively.

`calib`

Contains description files for the calibration plates, which you can use to calibrate your camera.

`doc/html/manuals`

Contains some of the User's Manuals.

`doc/html/reference/acquisition`

Contains the HTML documentation of the HALCON Image Acquisition Interfaces.

`doc/html/reference/io`

Contains the HTML documentation of the I/O device interfaces.

`doc/html/reference/operators`

Contains the HALCON Operator Reference.

`doc/index`

Contains the index files that are relevant for the full-text search of the documentation.

`doc/macros`

Contains the style files for the HALCON Operator Manual in all translated languages. These files set the layout and provide macros for names and formatting.

`doc/pdf/manuals`

Contains the PDF version of the User's Manuals.

`doc/pdf/reference`

Contains the PDF version of the HALCON Operator Reference in HDevelop syntax.

`doc/pdf/solution_guide`

Contains the PDF version of the Solution Guide I, the Solution Guide II, and Solution Guide III.

`filter`

Contains predefined filter masks.

`genicam`

Contains the underlying GenAPI runtime software for the GigEVision, GenICamTL, and USB3Vision image acquisition interfaces.

`help`

The files in this directory act as the HALCON database, i.e., they provide information about HALCON to HDevelop and to all HALCON applications. In particular, they contain the operator database. The XML files starting with `index_manuals` contain the index data of the manuals. The XML files starting with `index_examples` contain the data for the `Browse Examples` dialog in HDevelop and those starting with `tip_of_the_day` contain the information for the Tip of the Day that appears in the HDevelop Start Dialog.

`include`

Contains the header files that are necessary to use HALCON within the programming languages C or C++.

`lib`

Contains the HALCON libraries and the libraries for the supported image acquisition interfaces.

`license`

The license file must be placed here (see section 4.1 on page 27).

`lut`

Contains predefined look-up tables.

`misc`
: Contains miscellaneous files for the installer, e.g., the GigE Vision filter driver.

`ocr`
: Contains pretrained fonts.

`procedures`
: Contains external procedures for HDevelop and HDevEngine.

A.1.1.3 macOS

HDevelop is located in `/Applications`.

The default location for macOS is `/Library/Frameworks/HALCON.framework`:

`calib`
: Contains description files for the calibration plates, which you can use to calibrate your camera.

`filter`
: Contains predefined filter masks.

`genicam`
: Contains the underlying GenAPI runtime software for the GigEVision, GenICamTL, and USB3Vision image acquisition interfaces.

`Headers`
: Contains the header files that are necessary to use HALCON within the programming languages C or C++.

`help`
: The files in this directory act as the HALCON database, i.e., they provide information about HALCON to HDevelop and to all HALCON applications. In particular, they contain the operator database. The XML files starting with `index_manuals` contain the index data of the manuals. The XML files starting with `index_examples` contain the data for the `Browse Examples` dialog in HDevelop and those starting with `tip_of_the_day` contain the information for the Tip of the Day that appears in the HDevelop Start Dialog.

`lut`
: Contains predefined look-up tables.

`misc`
: Contains miscellaneous files for the installer, e.g., the GigE Vision filter driver.

`ocr`
: Contains pretrained fonts.

`procedures`
: Contains external procedures for HDevelop and HDevEngine.

The following directories reside in `/Library/Application Support/HALCON-17.12`:

`bin`
: Contains some HALCON tools (`hcomp, hbench, hcheck_parallel`) and the `LicenseInstaller`.

`doc/html/manuals`
: Contains some of the User's Manuals.

`doc/html/reference/acquisition`
: Contains the HTML documentation of the HALCON Image Acquisition Interfaces.

`doc/html/reference/io`
: Contains the HTML documentation of the I/O device interfaces.

`doc/html/reference/operators`
Contains the HALCON Operator Reference.

`doc/index`
Contains the index files that are relevant for the full-text search of the documentation.

`doc/macros`
Contains the style files for the HALCON Operator Manual in all translated languages. These files set the layout and provide macros for names and formatting.

`doc/pdf/manuals`
Contains the PDF version of the User's Manuals.

`doc/pdf/reference`
Contains the PDF version of the HALCON Operator Reference in HDevelop syntax.

`doc/pdf/solution_guide`
Contains the PDF version of the Solution Guide I, the Solution Guide II, and Solution Guide III.

`license`
The license file must be placed here (see section 4.1 on page 27).

Additionally, `/Library/Frameworks` contains the HALCON libraries. The corresponding directories reside in subdirectories, e.g., in `/Library/Frameworks/HALCONCpp.framework` for C++.

A.1.2 Example Programs, Images, Models

The subdirectories of the directory designated by the environment variable `HALCONEXAMPLES` contain example programs for the different parts of the HALCON system. The default location is operating system-dependent:

Windows	`C:\Users\Public\Documents\MVTec\HALCON-17.12`
Linux	`/opt/halcon/examples`
macOS	`/Users/Shared/Library/Application Support/HALCON-17.12/examples`

If `%HALCONEXAMPLES%` is not set, `%HALCONROOT%\examples` will be used as a fallback.

- `3d_models`: This directory contains 3D object data used by the example programs.
- `c`: Examples for using HALCON within the programming language C (see also the Programmer's Guide, chapter 15 on page 99).
- `c#`: Examples for using HALCON within the programming language C# (see also the Programmer's Guide, section 11.1 on page 77).
- `cpp`: Examples for using HALCON within the programming language C++ (see also the Programmer's Guide, chapter 7 on page 49).
- `cpp.net`: Examples for using HALCON within managed C++ (see also the Programmer's Guide, section 11.1 on page 77).
- `extension_package`: The example user package `halconuser` (see also the Extension Package Programmer's Manual).
- `hdevelop`: Examples for using HALCON in HDevelop:
  - ▷ `Applications`: Examples that show how to realize machine vision applications.
  - ▷ `1D-Measuring...XLD`: Examples for the HALCON operators, in subdirectories following the operator hierarchy as in the HALCON Operator Reference or in HDevelop's menu `Operators`.
- `hdevengine`: Examples for using HDevEngine.
- `ia_integration`: Example for an image acquisition interface (see also the Image Acquisition Interface Programmer's Manual).

- images: This directory contains example images and, in subdirectories, image sequences. These images are used by the example programs described above.
- io_integration: Example for the integration of an I/O device interface.
- solution_guide: Examples referenced in the Solution Guide I, the Solution Guide II, and the Solution Guide III.
- vb.net: Examples for using HALCON within Visual Basic .NET (see also the Programmer's Guide, section 11.1 on page 77).

To experiment with the examples without modifying the distributed versions, you can create a private copy in your own working directory. Note, however, that the .NET examples reference the HALCON/.NET assembly with a local path, thus after copying you must restore the reference (see the Programmer's Guide, page 60). You can also modify the environment variable HALCONEXAMPLES to point to the new location.

A.1.3 Machine Configuration Data

The following directory contains machine-specific configuration data, e.g., results of the utility hcheck_parallel or the operator optimize_aop.

Windows	C:\ProgramData\MVTec\HALCON-17.12 (%ProgramData%...)
Linux	/opt/halcon ($HALCONROOT)
macOS	/Library/Frameworks/HALCON.framework

A.1.4 User Configuration Data

The following directory contains user-specific configuration data, e.g., HDevelop preferences (HDevelop.ini).

Windows	C:\Users\%USERNAME%\AppData\Roaming\MVTec\HALCON-17.12 (%AppData%...)
Linux	$HOME/.hdevelop/MVTec
macOS	$HOME/Library/Preferences/com.mvtec.HDevelop.plist

In addition, the following directory is created by Windows 7 when the user copies files into the installation folder without administrator privileges:

Windows	C:\Users\%USERNAME%\AppData\Local\VirtualStore\Program Files\MVTec\HALCON-17.12 (%AppData%...)

A.2 HALCON's Environment Variables

Most of the configuration necessary to work with HALCON amounts to setting environment variables, e.g., to tell HALCON the directories where to find images or extension packages etc. These environment variables are described below, after some information regarding the different platforms.

A.2.1 Setting Environment Variables Under Windows

The installation program setup.exe automatically sets the necessary environment variables, e.g., HALCONROOT, HALCONEXAMPLES, HALCONIMAGES, and PATH (see below). To take a look at these settings, open the dialog Start ▷ Settings ▷ Control Panel ▷ System (Windows Vista: ... ▷ System ▷ Advanced System Settings) and select Environment. Under Windows 7/Windows 8, search for "Environment" using the system search, and select the search result "Edit the system environment variables" which appears in the category "Control Panel" (Windows 7) or "Settings" (Windows 8). You can add or modify a variable by entering the name of a variable and the desired value. If a value consists of multiple items, e.g., the variable PATH, which may contain multiple directories, those items must be separated by *semicolons*. Please note that in order to modify variables set during the HALCON installation you need administrator privileges!

A.2.2 Setting Environment Variables Under Linux

As described in section 2.2.2 on page 14, you must set the necessary environment variables in a login script or a shell resource script.

A.2.3 HALCON-Specific Environment Variables

- `HALCONROOT`
  This is the most important environment variable. It designates the directory where HALCON is installed. A typical path is, for example, `C:\Program Files\MVTec\HALCON` (Windows) or `/opt/halcon` (Linux).

  If this variable is unset at the time HDevelop is run, or when the HALCON library is loaded, its value will be inferred from the path the executable or the library resides in, respectively. From this path the trailing part `bin\%HALCONARCH%` or `lib\%HALCONARCH%` will be removed. The variable `HALCONROOT` will then be set to the resulting path temporarily.

  Based on this variable, the system switches to subdirectories, which are important for running HALCON. Some of them are listed below; the HALCON file structure is described in section A.1 on page 37.

  - `%HALCONROOT%\help`
    The files in this directory act as the HALCON information database (see section A.1 on page 37 for more information).
  - `%HALCONROOT%\doc\html\reference\operators`
    HDevelop expects the HTML files of the operator reference in this directory.
  - `%HALCONROOT%\license`
    This directory contains the *license file* necessary for using HALCON (see chapter 4 on page 27).
  - `%HALCONROOT%\examples`
    If the variable `HALCONEXAMPLES` (see below) is not set, the system looks for example programs in this directory.
  - `%HALCONEXAMPLES%\images`
    If the variable `HALCONIMAGES` (see below) is not set, the system looks for image files in this directory.

- `HALCONEXAMPLES`
  This environment variable designates the directory where HALCON example programs are installed.

- `HALCONIMAGES`
  The system uses this environment variable to search for image files specified by a relative path. As a rule it contains several directory names, separated by semicolons (Windows) or colons (Linux).

- `HALCONARCH`
  This variable designates the used platform. More details can be found in section 1.3 on page 8.

- `HALCONEXTENSIONS`
  This is a list of directories in which user-defined extension operators (so-called *extension packages*) are kept. Each package consists of a number of operators linked into a shared library, plus the additional operator documentation in help files and HTML files. See section 2.12 on page 22 for information on how to install an extension package, and the Extension Package Programmer's Manual for details on creating your own extension packages.

- `HALCONSPY`
  If this environment variable is defined (regardless of the value) *before you start* a HALCON program, the HALCON debugging tool *HALCON Spy* is activated. This corresponds to call the HALCON operator `set_spy` with the parameters `"mode","on"` *within* a HALCON program. The main difference between the two modes for activating HALCON Spy is that by defining `HALCONSPY` it is possible to monitor an already linked HALCON program during runtime without modifications. For further information on how to use HALCON Spy and how to parameterize it via this environment variable please refer to the Programmer's Guide, section 3.1 on page 23.

- `HALCON_LICENSE_FILE`
  This environment variable can be set to the full path of a specific license file. In this case, the given license file is used and no further search is performed. Usually, the license file is searched in the current working directory and the installed `license` directory.

A.2.4 General Environment Variables

- PATH
  Windows: During the installation, the directory %HALCONROOT%\bin\%HALCONARCH% is automatically added to the system variable PATH.

  Linux: If you want to start HDevelop from an arbitrary directory, you must include the HALCON program path $HALCONROOT/bin/$HALCONARCH in the system variable PATH.

- LD_LIBRARY_PATH (Linux only)
  Please include the HALCON library path $HALCONROOT/lib/$HALCONARCH in the system variable LD_LIBRARY_PATH. This is necessary both for running HDevelop and for creating stand-alone applications.

- DISPLAY (Linux only)
  The system uses this environment variable to open windows. It is used in the same way as for other X applications.

- HOME (Linux only)
  This system variable points to your home directory.

A.3 Registry Keys on Windows Sytems

The Windows installer adds several keys to the system registry. The following sections list these registry keys. If multiple versions of HALCON are installed, version-specific information is stored under HKLM\SOFTWARE\MVTec\HALCON\version, where *version* is the major version number and, optionally, the edition, e.g., 17.12-Progress.

! Please note: On **x64-based systems** a different path is used: HKLM\SOFTWARE\Wow6432Node\MVTec\...

A.3.1 Registered File Types

```
HKCR\.hdev
HKCR\HDevelop.Source.File
HKCR\.dev
HKCR\.hdvp
HKCR\HDevelop.External.Procedure
HKCR\.dvp
HKCR\HDevelop.Procedure.Library
HKCR\.hdpl
HKCR\HALCON.Object
HKCR\.hobj
```

A.3.2 Environment Variables

```
HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\
HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\HALCONARCH
HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\HALCONEXAMPLES
HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\HALCONIMAGES
HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\HALCONROOT
HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\PATH (modified)
```

A.3.3 Basic Installation Data

```
HKLM\SOFTWARE\MVTec\HALCON\17.12-Progress\Initialized\InstallDir
HKLM\SOFTWARE\MVTec\HALCON\17.12-Progress\Initialized\ProgramGroup
HKLM\SOFTWARE\MVTec\HALCON\17.12-Progress\Initialized\Architecture
HKLM\SOFTWARE\MVTec\HALCON\17.12-Progress\Initialized\ExampleRoot
HKLM\SOFTWARE\MVTec\HALCON\17.12-Progress\Initialized\LanguageDoc
HKLM\SOFTWARE\MVTec\HALCON\17.12-Progress\Initialized\InstalledVersion
HKLM\SOFTWARE\MVTec\HALCON\17.12-Progress\Initialized\InstalledRevision
HKLM\SOFTWARE\MVTec\HALCON\17.12-Progress\Initialized\InstalledBuild
```

Sections selected during installation:

```
HKLM\SOFTWARE\MVTec\HALCON\17.12-Progress\InstalledComponents
```

(depending on the sections)

A.3.4 Uninstall Information

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\InstallLocation
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\UninstallString
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\DisplayIcon
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\DisplayName
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\DisplayVersion
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\Publisher
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\Version
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\URLInfoAbout
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\URLUpdateInfo
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\NoModify
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 17.12 Progress\NoRepair
```

Index

compatibility (HALCON), 8
cuBLAS library, 25
cuDNN library, 25
customize HALCON installation, 18

Deep Learning installation Linux, 16
Deep Learning installation Windows, 13
Deep Learning third-party components, 25
development license, 29
development version, 7
DISPLAY (environment variable), 44

environment variables (general), 44
environment variables (HALCON), 43
evaluation license, 28
examples (HALCON directory), 41

get host ID manually, 28
get host ID of network card, 29
get required HALCON modules, 30

HALCON_LICENSE_FILE (environment variable), 43
HALCONARCH (environment variable), 43
HALCONEXAMPLES (environment variable), 43
HALCONEXTENSIONS (environment variable), 43
HALCONIMAGES (environment variable), 43
HALCONROOT (environment variable), 43
HALCONSPY (environment variable), 43
hhostid, 28, 34
HOME (environment variable), 44

images (HALCON directory), 42
install additional HALCON parts, 17
install extension package, 22
install HALCON, 11
install HALCON for the first time, 11
install image acquisition interface, 21
install runtime version, 17
install runtime version (silent), 14
installation troubleshooting, 33
installed file structure, 37

LD_LIBRARY_PATH (environment variable), 44
license, 27
  overview, 10
license file, 27
licensing troubleshooting, 34

maintenance release, 7

nodelocked development license bound to dongle, 29
nodelocked development license bound to network card, 29

PATH (environment variable), 44

runtime license, 30
runtime version, 7

set environment variables
  Linux, 43
  Windows, 42
switch between HALCON platform versions, 21
switch between HALCON versions, 21
system requirements, 8

troubleshooting (miscellaneous), 35
troubleshooting for get host ID, 34

uninstall HALCON, 19
uninstallation troubleshooting, 33
update HALCON, 18
upgrade HALCON, 18
upgrade license, 31

version (HALCON), 7

WOW64, 9