HALCON

the power of machine vision

Installation Guide

MVTec Software GmbH

Building Vision for Business

All about installing and licensing HALCON, Version 12.0

Edition 1	December 2003	(HALCON 7.0)
Edition 1a	July 2004	(HALCON 7.0.1)
Edition 1b	April 2005	(HALCON 7.0.2)
Edition 2	July 2005	(HALCON 7.1)
Edition 2a	April 2006	(HALCON 7.1.1)
Edition 2b	December 2006	(HALCON 7.1.2)
Edition 3	June 2007	(HALCON 8.0)
Edition 3a	October 2007	(HALCON 8.0.1)
Edition 3b	April 2008	(HALCON 8.0.2)
Edition 4	December 2008	(HALCON 9.0)
Edition 4a	June 2009	(HALCON 9.0.1)
Edition 4b	March 2010	(HALCON 9.0.2)
Edition 5	October 2010	(HALCON 10.0)
Edition 5a	April 2011	(HALCON 10.0.1)
Edition 6	May 2012	(HALCON 11.0)
Edition 6a	November 2012	(HALCON 11.0.1)
Edition 6b	July 2013	(HALCON 11.0.2)
Edition 6c	April 2014	(HALCON 11.0.3)
Edition 7	November 2014	(HALCON 12.0)

Protected by the following patents: US 7,062,093, US 7,239,929, US 7,751,625, US 7,953,290, US 7,953,291, US 8,260,059, US 8,379,014, US 8,830,229. Further patents pending.

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 10, which gives an overview of the available documentation. ☜

Notation

Except for Linux/OS X-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 65 for more information on environment variables). The same expression in Linux/OS X convention would look like

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

Contents

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 13 and chapter 3 on page 31, 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++, .NET, and COM, interfaces to more than 50 frame grabbers, I/O device interfaces and hundreds of industrial cameras, 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 12.0 or HALCON 11.0, to differentiate them from so-called *maintenance releases* like HALCON 11.0.1. 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 of existing operators.

- **Compatibility**
  A new HALCON *version* is not downward compatible, with the following implications: First, you must upgrade your HALCON license (see section 3.5 on page 43). 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 System Requirements

You can use HALCON under Windows, Linux, and OS X. The summary of system requirements is listed in table 1.1; more details follow below.

Operating System	Processor	Compiler / Environment
Windows	Intel Pentium 4 / AMD Athlon 64 or higher	Visual Studio 2005 or higher
Windows x64	Intel 64 or AMD64	Visual Studio 2005 or higher
Linux x86_64	Intel 64 or AMD64	`gcc 4.2` or higher
OS X	Intel 64	Xcode

Table 1.1: Platforms supported by HALCON.

Additional Linux Requirements

The Linux distribution has to be LSB compliant. The corresponding packages have to be installed, e.g., redhat-lsb (Fedora, RedHat), lsb (SuSE), lsb-base + lsb-core (Ubuntu).

Furthermore, an XServer has to be installed. This is required even for command-line tools provided with HALCON.

Platform-Specific HALCON Versions

For each of the operating systems listed in table 1.1, platform-specific versions of HALCON's executables and libraries are provided. Table 1.2 lists all platform-specific versions with detailed system requirements. 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 OS X, 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.

Further note that in order to create .NET applications under Linux/OS X you need to install Mono.

HALCONARCH	**Operating System, Processor**	**Compiler**
x86sse2-win32	Windows Vista/7/8 or Windows Server 2008/2012, on x86 processor with SSE2 extension, e.g., Intel Pentium 4 / AMD Athlon 64 or higher	Visual Studio 2005 or higher
x64-win64	Windows Vista/7/8 or Windows Server 2008/2012 x64 Edition, on Intel 64 or AMD64	Visual Studio 2005 or higher
x64-linux	Linux x86_64, Kernel 2.6 or higher, libc.so.6 (GLIBC_2.6 or higher), libstdc++.so.6 (GLIBCXX_3.4 or higher), on Intel 64 or AMD64	gcc 4.2 or higher
x64-macosx	OS X 10.8 or higher on Intel 64	Xcode

Table 1.2: Values of HALCONARCH and detailed system requirements.

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.

1.4 Licensing

To run HALCON on a computer, you need a license. The license itself is typically stored in a file named "license.dat" or similar.

Licenses are always issued for a certain HALCON version (i.e., major release, see section 1.2 on page 8), e.g., for HALCON 11.0. However, a license is not exclusively bound to this version: It is *upward compatible within the version number*, i.e., licenses for HALCON 11.0 are also valid for HALCON 11.0.1.

If you want to use different HALCON versions at the same time, from version 7.1 on the name of the license file is not restricted to "license.dat" anymore. Now, it has to begin with "license" and end with ".dat", but in between other information can be included. Thus, different licenses for different versions (e.g., license-12.0.dat for version 12.0) 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 3 on page 31.

- **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 3.1 on page 31).

  If you want to use HALCON on multiple computers simultaneously, you need a license for each of them. You can either obtain multiple *node-locked licenses*, or, if the computers are connected via a network, you can also use a *floating license*. The main point of using floating licenses is that you do not need to specify on which computers you want to run HALCON, but only on how many of them simultaneously. Floating licenses are described in detail in section 3.3.3 on page 38.

- **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); note, however, that there are no floating runtime licenses.

  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), and install it for the first time (section 2.2),
- install additional parts (section 2.3 on page 20),
- manually install a runtime version of HALCON (section 2.5 on page 21),
- update HALCON to a newer maintenance release (section 2.6 on page 22), and
- upgrade HALCON to a newer version (section 2.7 on page 22).

Furthermore, this chapter explains how to

- uninstall HALCON (section 2.9 on page 23),
- manage multiple versions in parallel (section 2.10 on page 26),
- switch between HALCON for 32-bit and 64-bit Windows or vice versa (section 2.11 on page 27),
- install updates of image acquisition interfaces (section 2.12 on page 27), and
- install extension packages (section 2.13 on page 27).

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). In addition to the installation package for your platform, it is highly recommended to download the corresponding package containing example images and 3D models. Most of the example programs will not run without these files. Apart from the instructions in the following sections, the installation procedure is described in detail on the web pages themselves.

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 17
- on OS X systems: section 2.2.3 on page 19

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

If you want to install only the license manager daemon, consider the option “Compact” in Step 3, or install it manually as described in section 4.3.3.1 on page 54.

2.2.1 Installing HALCON under Windows

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

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 9). 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.

If you are using ActivVisionTools, the setup program checks whether the installed version is compatible to the current HALCON version.

By default, the installer also checks for available maintenance releases. You can disable this check, e.g., if the computer is not connected to the internet.

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 20).

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 61 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.

This installation type is also recommended if all you want to do is install the license manager daemon for floating licenses but prefer not to do it manually. If HALCON will never be used on the floating license server and disk space does not permit the extra overhead of a minimal HALCON installation, the recommended method is to install the license manager daemon manually. This is described in section 4.3.3 on page 53.

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 concerning licensing

After this selection, the setup program asks whether to install the driver programs that are necessary if you want to use a dongle-bound license. Section 3.3.2 on page 37 provides detailed information about these dongle drivers, including how to install them manually if you do not let the setup program install them.

The dialog also asks whether the computer is to act as the so-called *floating license server*. In this case, the setup program installs the so-called *license manager daemon*, a system service that manages your floating licenses. Note that when using floating licenses, you need to install the license manager daemon only on one computer: the license server. Please refer to section 3.3.3 on page 38 for more information about floating licenses, e.g., how to install the license manager daemon manually.

Step 4: 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 MVTec GigE Vision interface because it increases the performance significantly.

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: Select installation destination

After these checks you can choose a folder where HALCON is to be installed, e.g., `C:\Program Files\MVTec\HALCON-12.0`.

Step 7: Example programs and images

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 8: The actual installation

Finally, the actual installation starts, which includes copying files to the destination folder (see section A.1 on page 61 for an overview of HALCON's fileset), entering information in the Windows registry, and setting environment variables (see also section A.2 on page 65). Depending on the installation type and content, it may be necessary to reboot the computer after the installation is finished.

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.

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 17. The optimization program may be started from the Windows start menu.

2.2.2 Installing HALCON under Linux

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

```
tar zxvf halcon-12.0-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 9). 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 14. 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 20). 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: Set environment variables

After the installation, the following environment variables must be set or modified in order for HALCON to work (see section A.2 on page 65 for more information about these and other 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}:${HALCONROOT}/FLEXlm/${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.

- `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 1.2 on page 10)
- `PATH`: this system variable should include `$HALCONROOT/bin/$HALCONARCH` and `$HALCONROOT/FLEXlm/$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. The suitable shell script can be used to 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 7: Further configuration

If you are using *floating licenses*, you must install and start the license manager daemon as described in section 4.3.3 on page 53.

If you want to use a **dongle-bound license**, see section 4.3.2.2 on page 53 for information how to install the USB dongle driver.

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 17.

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

2.2.3 Installing HALCON under OS X

To install HALCON on OS X 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-12.0/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 61 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-12.0/license`.

Step 6: Further configuration

If you are using *floating licenses*, you must install and start the license manager daemon as described in section 4.3.3 on page 53.

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 17.

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.

OS X systems

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

2.4 Usage of the command `regsvr32` on Windows x64

Whenever the command `regsvr32` is used in this manual, you have to make sure to use the correct version of the binary on Windows x64.

If you are installing/uninstalling the 32-bit version of HALCON on Windows x64, use

```
C:\WINDOWS\SysWOW64\regsvr32
```

If you are installing/uninstalling the 64-bit version of HALCON on Windows x64, use

```
C:\WINDOWS\system32\regsvr32
```

2.5 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 2.1 on page 13 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 65.

2. Copy the required runtime DLLs to, e.g., C:\%HALCONROOT%\bin\%HALCONARCH% (see table 1.2 on page 10 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, e.g., hAcq1394IIDC.dll, 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 16.1 on page 128).

   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. Register halconx.dll (or halconxxl.dll) if the application uses the HALCON/COM interface. To do this, execute regsvr32 halconx.dll in the directory where the file resides (Windows x64 users: note section 2.4 on page 20).

5. Register hdevenginex.dll (or hdevenginexxl.dll) if the application uses HDevEngine. To do this, execute regsvr32 hdevenginex.dll in the directory where the file resides (Windows x64 users: note section 2.4 on page 20).

6. Copy the HALCON help files to %HALCONROOT%\help.

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

8. If you are using dongle-based licensing, the corresponding dongle driver needs to be installed as well (see section 4.3.2.1 on page 52).

9. Depending on the application, one or more of the following directories need to be copied to `%HALCONROOT%` as well: `calib`, `filter`, `lut`, `ocr`, `procedures`.
10. If you intend to use the GigEVision, GenICamTL, or USB3 Vision interface, copy the directory `genicam` to `%HALCONROOT%` as well.

2.6 Updating HALCON

With HALCON, the term “update” means to install a newer maintenance release over a release based on the same HALCON version, e.g., HALCON 12.0.1 over 12.0. As described in section 1.2 on page 8, 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.

2.7 Upgrading HALCON

With HALCON, the term “upgrade” means to install a newer version over an older one, e.g., HALCON 12.0 over 11.0 or 11.0.1. As described in section 1.2 on page 8, 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 2.9. If you want to keep the older version, please refer to section 2.10 on page 26.

The installation process itself is the same as the first-time installation described in section 2.2 on page 14. Please note that after installing the new version you must upgrade the license as described in section 3.5 on page 43.

2.8 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. It can be downloaded from `http://www.halcon.com/download`.

2.9 Uninstalling HALCON

2.9.1 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 2.9.1.2.

2.9.1.1 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 12.0` ▷ `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 4.2 on page 46.

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 12.0\.aop_info`. You may remove these files manually without risk. Furthermore, the uninstaller does not remove the dongle drivers as they might be needed by another application; section 3.3.2 on page 37 describes how to remove them manually. Moreover, the uninstaller does not remove the MVTec GigE Vision streaming filter driver if it is installed. This driver needs to be removed separately.

2.9.1.2 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. Unregister the HALCON/COM interface and the COM version of HDevEngine (Windows x64 users: note section 2.4 on page 20). If you have been using HALCON XL, append the suffix `xl` to the `.dll` files in the following commands (thus, `halconx.dll` becomes `halconxxl.dll`, for example).

   ```
   regsvr32 /u "%HALCONROOT%\bin\%HALCONARCH%\halconx.dll"
   regsvr32 /u "%HALCONROOT%\bin\%HALCONARCH%\hdevenginex.dll"
   ```

2. For extensive cleaning of the Windows registry you can optionally run the program `%HALCONROOT%\misc\x86-win32\clean_reg_halconx.exe`.

3. 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%"
   ```

4. 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)
   ```

5. 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 65 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
   ```

6. 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.

2.9.2 Uninstalling HALCON under Linux

HALCON provides no uninstallation script for Linux systems, therefore you must perform the uninstallation manually. In case that you are using a floating license you have to uninstall the floating license daemon as well (see section 3.3.3 on page 38).

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 65); 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 17).

To remove the dongle driver, refer to section 4.3.2.2 on page 53.

2.9.3 Uninstalling HALCON under OS X

To uninstall HALCON from OS X 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. It does not remove the property list file launched to start the floating license manager (if you use floating licenses; see section 4.3.3.3 on page 57), nor does it remove the dongle driver /usr/bin/hasp_darwin.dylib if it was installed. To remove the dongle driver, refer to section 4.3.2.2 on page 53.

The script will also 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
```

2.10 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 18 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 65 for more information about setting environment variables.

If you are using the HALCON/COM interface or the COM version of HDevEngine, you must also reregister the corresponding DLLs halconx.dll or hdevenginex.dll, for example as follows: Open a Windows Command Prompt and change into the subdirectory bin\%HALCONARCH% of the root directory of your “old” HALCON installation (please note that versions older than HALCON 8.0 did not set the environment variable HALCONARCH and used different names for the platform-specific subdirectories!). Unregister halconx.dll or hdevenginex.dll (Windows x64 users: note section 2.4 on page 20):

```
regsvr32 /u halconx.dll
regsvr32 /u hdevenginex.dll
```

Then change into the corresponding subdirectory of your other HALCON installation and register its halconx.dll or hdevenginex.dll by typing

```
regsvr32 halconx.dll
regsvr32 hdevenginex.dll
```

With the same method, re-register hdevenginex.dll, i.e., the DLL of the COM version of HDevEngine.

If you are using HALCON/COM XL, you must re-register halconxxl.dll and hdevenginexxl.dll instead.

As an alternative to the Command Prompt, you can unregister and register halconx.dll or hdevenginex.dll via the dialog Start ▷ Run together with the Windows Explorer: In the latter, "open" the directory bin\%HALCONARCH% of the root directory of your old HALCON installation. Now, type regsvr32 /u in the dialog Run and then drag and drop halconx.dll or hdevenginex.dll from the Windows Explorer into the dialog, where it automatically appears with the full path. To execute the command, click OK. Then, open the directory bin\%HALCONARCH% of the root directory of the other HALCON installation in the Windows Explorer, type regsvr32 in the dialog Run, drag and drop halconx.dll or hdevenginex.dll from the Windows Explorer into the dialog, and again click OK.

2.11 Switching between Different HALCON Platform Versions under Windows

As described in section 1.3 on page 9, HALCON is provided for different platforms. However, only the first installation sets the environment variable PATH (see section A.2 on page 65). 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 1.2 on page 10), which is referenced in PATH.

2.12 Installing HALCON Image Acquisition Interfaces

With every HALCON installation, you automatically obtain the latest release of the currently available image acquisition interfaces (see also section A.1 on page 61). In between HALCON releases, however, 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, and install it as described on the corresponding web pages.

2.13 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/OS X 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 65) to access those images without specifying a complete path.

2.13.1 Using an Extension Package Within HDevelop

In order to use a new package within HDevelop under Windows/OS X, 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 1.2 on page 10 for the possible values of HALCONARCH).

2.13.2 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/OS X) to the application code, in addition to the HALCON library.

2.13.2.1 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 1.2 on page 10 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!

2.13.2.2 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.

2.13.2.3 Using an Extension Package Under OS X

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

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 3.2 on page 35),
- development licenses (section 3.3 on page 36), and
- runtime licenses (section 3.4 on page 41),

Finally, section 3.5 on page 43 shows how to upgrade a license.

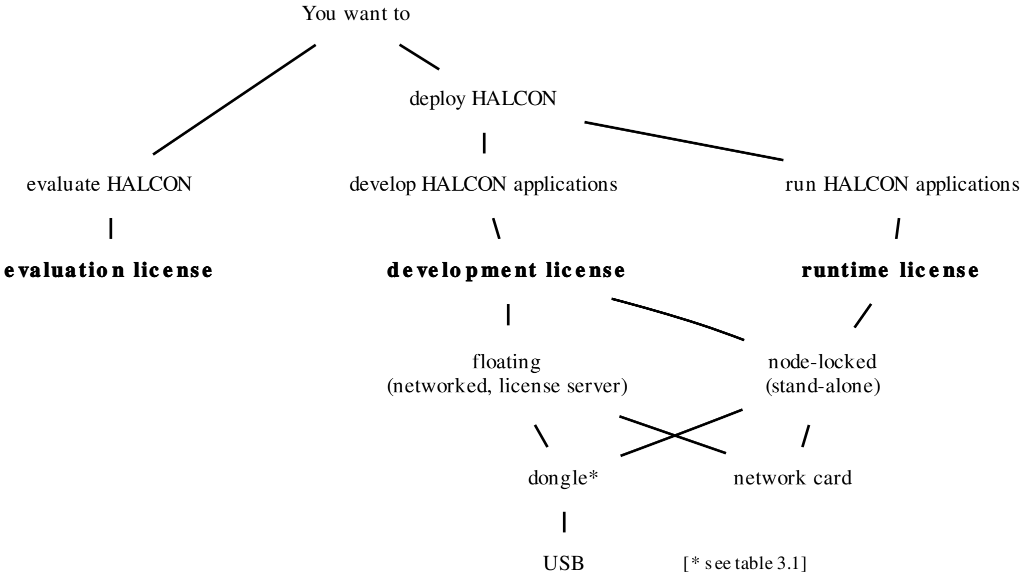
3.1 What is a License?

HALCON's licensing mechanisms are based on the license manager software *FlexNet Publisher* (formerly named *FLEXlm*) from Flexera Software. The licenses themselves are stored in so-called *license files*; example files are depicted in the following sections. The content of these files specifies

- what is licensed (e.g., development version, runtime version, etc.)
- whether the license is temporary (evaluation license) or permanent
- the hardware to which the license is bound (see below)
- additional information for floating licenses.

License files are named `license.dat` (or `license-12.0.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!** !

Available Licenses



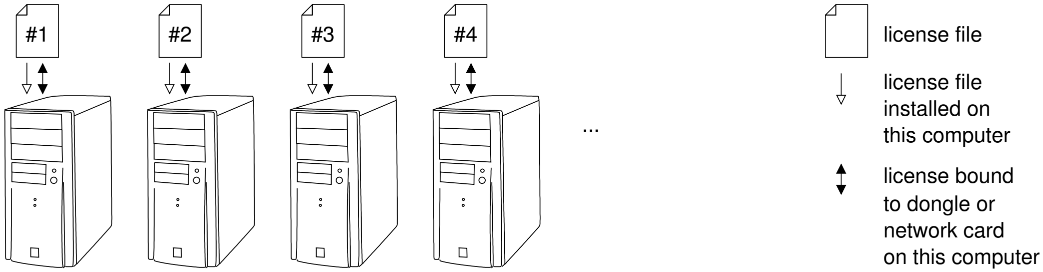


Figure 3.1: Individual node-locked licenses.

Node-Locked Versus Floating Development Licenses

You can choose between two types of development licenses. They differ in the number of computers that can run HALCON applications simultaneously and in their method of license validation.

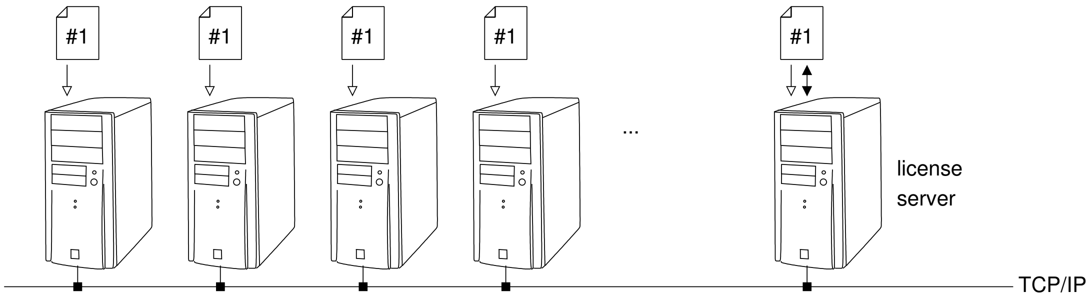


Figure 3.2: Floating license: License server approves remote HALCON instances.

Firstly, there are *node-locked licenses*. Such a license allows to run exactly one installation of HALCON at a time. The license validation is done on the local computer. To have additional people use HALCON on different computers at the same time, additional node-locked licenses have to be purchased (see figure 3.1). However, node-locked licenses can travel between computers by moving the hardware they are bound to (see below).

Then, there are *floating licenses* which require participating computers to be networked. When ordering a floating license, you have to specify the number of concurrently running HALCON instances (see page 38) and decide on a computer in your network to act as a designated *license server*. There is no need to specify the participating computers. When the ordered license file arrives it is copied to the license server and each HALCON installation. The license validation is performed on the license server, which monitors concurrently running instances of HALCON in the network (see figure 3.2).

In contrast to development licenses, runtime licenses are always node-locked.

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 3.3.1 on page 36) or a *dongle* (see section 3.3.2 on page 37 and table 3.1).

Dongle-bound node-locked 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 node-locked license on Windows and Linux systems.

If you are developing applications based on HALCON in a team on different computers in a network, you can use a floating license. This license is bound to the network card or dongle which is attached to the license server. The license server may be changed by moving the dongle and altering the server entry in the license file. Floating licenses also are an option if a single person wants to use HALCON on different computers within a network at different times. For more details about floating licenses see section 3.3.3 on page 38.

Hardware Binding of the License on the Supported Platforms

Consult table 3.1 to find out which hardware is supported for binding your license to on your operating system.

	License can be bound to	
Operating system	**network card**	**USB port dongle**
Windows	yes	yes
Windows x64	yes	yes
Linux x64	yes	yes
OS X	yes	yes

Table 3.1: License binding options on the supported platforms.

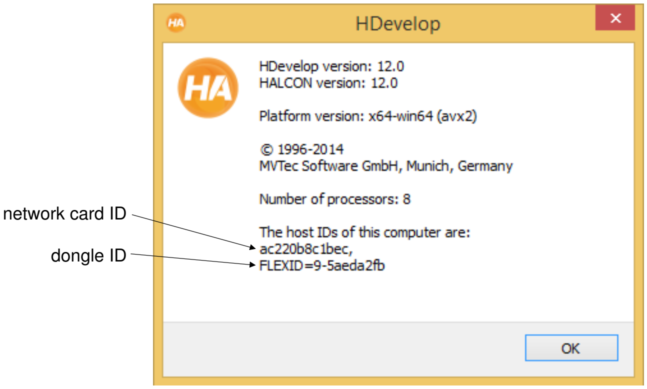
Identifying the Hardware

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

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

`Help ▷ About.`

For an example see figure 3.3, which was generated on a computer equipped with a network card and a dongle.



HDevelop version: 12.0
HALCON version: 12.0

Platform version: x64-win64 (avx2)

© 1996-2014
MVTec Software GmbH, Munich, Germany

Number of processors: 8

The host IDs of this computer are:
ac220b8c1bec,
FLEXID=9-5aeda2fb

Figure 3.3: Identifying information in the HDevelop window `About`.

Alternatively, you can execute the following commands from a Windows Command Prompt or a Linux/OS X shell to get the host ID and the dongle ID, respectively. See section 4.3.1 on page 51 for a detailed description on solving problems extracting the host ID. The utility `lmhostid` is located in `%HALCONROOT%\FLEXlm\%HALCONARCH%` (see table 1.2 on page 10 for the possible values of `HALCONARCH`).

```
lmhostid -ether
lmhostid -flexid
```

You can safely ignore the warning 'WIBU DLL not found' which will be returned. If these commands fail, HALCON will not be able to validate the license file.

3.2 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. The distributor will send you a *license file* that looks similar to the one depicted in figure 3.4: The lines starting with `#` are comments; the date `03/2015` indicates that the evaluation license is valid until the end of March 2015.

```
##############################################
# Evaluation License for 03/2015  (ID: DEMO) #
##############################################
FEATURE MVTec_HALCON mvtecd 12.0 01-mar-2015 uncounted VENDOR_STRING=511 \
  HOSTID=DEMO SIGN="DEAD BEEF 66E3 BF26 3AFD 1860 FB2E 7C50 A0EE F2EE \
  A324 E9EF FF57 9DD0 4EA7 094A 1D3D 19C2 1678 F817 327E 1DC2 004C \
  8825 2D5E C7A8 A397 5F91 7A5E 98E2"
FEATURE MVTec_HDevelop mvtecd 12.0 01-mar-2015 uncounted VENDOR_STRING=511 \
  HOSTID=DEMO SIGN="DEAD BEEF 3674 9B07 9E53 1894 681E E3A0 5275 A60A \
  78B2 2304 A6AC BB4A 87C1 090F B47E 70D2 3231 C571 BA53 FB84 D9A0 \
  4AE6 3A7A 20C5 9F24 A737 8A3D A27C"
```

Figure 3.4: Example evaluation license.

The lines starting with `FEATURE` contain the actual license data; they are called *license keys*. Evaluation licenses contain two license keys: `MVTec_HALCON` includes those HALCON parts that are necessary for running HALCON, while `MVTec_HDevelop` includes additional parts for developing, e.g., HDevelop.

As you can see, the license keys contain the licensed version number of HALCON (figure 3.4: `12.0`). As noted in section 1.4 on page 10, the license is upward compatible within the version number, i.e., licenses for HALCON 12.0 are also valid for all maintenance releases of HALCON 12.0, e.g., HALCON 12.0.1.

The entry following the HALCON version specifies when the evaluation license key expires (figure 3.4: `01-mar-2015`).

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-12.0.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.

3.3 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). But in contrast to the evaluation license, a development license is permanent, i.e., there is no temporal restriction. However, it must be bound to a certain hardware component (see also section 3.1 on page 31). The following sections describe how to proceed to obtain and install a

- node-locked license bound to a network card (section 3.3.1)
- node-locked license bound to a dongle (section 3.3.2)
- floating license bound to a network card or dongle (section 3.3.3 on page 38)

3.3.1 Node-locked License Bound to a Network Card

Step 1: Extract the host ID

As already described in section 3.1 on page 34, the easiest way to extract the host ID is to execute the following command from a Windows Command Prompt or a Linux/OS X shell.

```
lmhostid -ether
```

Alternatively, start HDevelop and select the menu item `Help` ▷ `About`. This dialog displays the host ID of the network card, e.g., the string `"00270e0ac34a"` in figure 3.3 on page 34.

If HDevelop fails to detect a host ID although your computer does have a network card, please try to extract the host ID manually as described in section 4.3.1 on page 51. This section also describes what to do if IDs like `"ffffffff"` or `"0"`, or multiple IDs are displayed.

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* that looks similar to the one depicted in figure 3.5.

Like the evaluation license depicted in figure 3.4 on page 35, it contains two *license keys* (lines starting with `FEATURE`), one for the runtime parts of HALCON (`MVTec_HALCON`) and one for the development parts (`MVTec_HDevelop`). The differences to the evaluation license show up as follows:

```
###########################################
# MVTec Software GmbH (ID:  00270e0ac34a) #
# Development License                     #
###########################################
FEATURE MVTec_HALCON mvtecd 12.0 01-jan-0000 uncounted VENDOR_STRING=511 \
  HOSTID=00270e0ac34a ONE_TS_OK SIGN="DEAD BEEF D9E1 BDCE 3B6F 1008 1062 \
  0FC1 0214 C5EB FG1B 9B3C 49C4 1CD3 DA0F 0F5D B870 36C4 D91A 2B21 \
  169F 26BC 5915 383E 71C7 153B 5440 5H4A 5458 5E1D"
FEATURE MVTec_HDevelop mvtecd 12.0 01-jan-0000 uncounted VENDOR_STRING=511 \
  HOSTID=00270e0ac34a ONE_TS_OK SIGN="DEAD BEEF 8D52 4CF4 A0CB D4A9 D443 \
  8B22 4DC8 6734 3190 A4D9 047A 7261 B123 06A1 D0A2 3012 6D8F 9E49 \
  1438 ECD9 3AF8 9978 7924 4E92 4D94 C248 0262 3FA1"
```

Figure 3.5: Example node-locked development license, bound to network card.

- The expiration date is set to 01-jan-0000, which means that the license is permanent (alternatively, the entry may contain the string permanent).
- Both license keys are bound to the HOSTID you extracted in the first step (in the example: 00270e0ac34a).

If the FEATURE line contains the additional keyword ONE_TS_OK, then this license allows also the checkout from a *terminal server* for a single user (Windows only).

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-12.0.dat or similar, see section 1.4 on page 10).

3.3.2 Node-locked License Bound to a Dongle

Dongles are available for selected platforms (see table 3.1 on page 34. Note that for Windows and OS X systems we assume that you let the setup program install the driver programs necessary for using dongles as described in section 2.2.1 on page 14. If you did not install the drivers, please refer to section 4.3.2 on page 52. On Linux systems, the dongle driver has to be installed manually, which is described in section 4.3.2.2 on page 53.

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*, which looks similar to the one depicted in figure 3.6. The only difference to the network card license depicted in figure 3.5 is that the entry HOSTID now contains the ID of the dongle (in the example: FLEXID=9-a6305af4). This ID is also printed on the back of the dongle. USB dongles have IDs starting with 'FLEXID=9-'.

```
###############################################
# MVTec Software GmbH (ID: FLEXID=9-a6305af4) #
# Development License                         #
###############################################
FEATURE MVTec_HALCON mvtecd 12.0 01-jan-0000 uncounted VENDOR_STRING=511 \
  HOSTID=FLEXID=9-a6305af4 ONE_TS_OK SIGN="DEAD BEEF 4BE3 30D0 F631 13D5 \
  D694 ED77 D4D8 2A35 AA31 6672 1651 EC07 C392 031E 197A CF39 005A \
  4811 6DE3 3BA5 0549 CA11 FE97 68C3 15F9 62E4 DA06 3E96"
FEATURE MVTec_HDevelop mvtecd 12.0 01-jan-0000 uncounted VENDOR_STRING=511 \
  HOSTID=FLEXID=9-a6305af4 ONE_TS_OK SIGN="DEAD BEEF 3C77 2773 63B0 79B6 \
  B1B7 4D56 C16E 749E D959 37DB DD11 EBA9 906B 0C6E 0E99 C20E 91E7 \
  9037 A47A 37A8 B010 84BE D518 480B 5318 EE81 DCE4 5A6D"
```

Figure 3.6: Example development license, bound to 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-12.0.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.

3.3.3 Floating License Bound to a Network Card or Dongle

Section 3.1 on page 31 already briefly described the basic concept of floating licenses; now, we take a closer look:

- With a floating license, you can use HALCON on multiple computers in a network without having to identify each of these computers.
- When ordering a floating license via your local distributor you have to specify *how many* HALCON instances are allowed to run simultaneously.

  To determine this number, add the number of users per computer:

  - single user runs *many* HALCON applications on the *same* computer: 1 instance,
  - single user runs HALCON applications on two computers: 2 instances,
  - two users run HALCON applications on the *same* computer: 2 instances,
  - two users on computer A, three users on computer B: 5 instances, and so forth...
- Only one computer must be identified: the *license server*. On this computer, the *license manager daemon* is installed, a program that keeps track of the HALCON applications currently being run (see section 4.3.3 on page 53 for details).
  As already described for non-floating licenses, the computer which acts as the license server can be identified via network card or dongle.
- Note that **only development licenses are available as floating licenses**.

Floating licenses are well-suited especially in the following scenarios:

- A single person wants to develop HALCON applications on different computers within a network, and does not want to use a dongle-bound license.
- A group of persons wants to develop HALCON applications simultaneously in a computer network.

Step 1: Choose the computer acting as the license server

In principle, any computer can be chosen as the license server; it need not be a “server” in the sense that it must provide special functionality or have a special kind of operating system, e.g., Windows 2008 Server, installed. The only requirement is that the computer must be accessible whenever HALCON is to be used in the network.

You can even use different architectures for the license server and for the HALCON applications, respectively, e.g., a Linux workstation for the license server and Windows for developing HALCON applications. The license server can also be used for developing HALCON applications; but even if not, HALCON must be installed on it.

Step 2: Extract the host ID of the license server

Like all development licenses, floating licenses must be bound to a hardware component. Here, it is the computer acting as the license server that must be identified, be it via a network card or a dongle. Please refer to section 3.3.1 on page 36 (network card). If you choose a dongle-bound license, no further action is required as you get the dongle together with the license (see section 3.3.2 on page 37).

Step 3: Obtain the license

Send the desired number of licenses (i.e., the maximum number of HALCON applications that should run simultaneously per user on different computers), the hostname of the computer which acts as the license server, and – except in case of a dongle-bound license – the extracted host ID of the license server to your local distributor.

The distributor then sends you a license file, which looks similar to the one depicted in figure 3.7. If you requested a dongle-bound license, you will also receive the dongle.

Step 4: Adapt the license

In contrast to the license types described in the previous sections, you may need to adapt parts of the floating license file. Therefore, we take a closer look at the example floating license depicted in figure 3.7. It is bound to the same network card as the example node-locked license depicted in figure 3.5 on page 37. If the license is bound to a USB dongle, make sure that the dongle ID is used instead of the host ID (see the following description).

```
#################################################
# MVTec Software GmbH (ID: 00270e0ac34a)        #
# Development License                           #
#################################################
SERVER myservername 00270e0ac34a 27000
VENDOR mvtecd "C:\Program Files\MVTec\HALCON-12.0\FLEXlm\x64-win64\mvtecd"
FEATURE MVTec_HALCON mvtecd 12.0 01-jan-0000 7 VENDOR_STRING=511 \
  DUP_GROUP=UH SIGN="DEAD BEEF 749D 6F3A 986E 93F9 754F EAFE 0B78 \
  B20A 9319 AFEF A7FC 9CAC B75C 049D 2ED5 F54F 3778 A8E5 6C61 4F01 \
  9C2A 84AB 1B2D 4D36 66A1 215C 6935 64E9"
FEATURE MVTec_HDevelop mvtecd 12.0 01-jan-0000 7 VENDOR_STRING=511 \
  DUP_GROUP=UH SIGN="DEAD BEEF B31A 7CB4 0195 73D9 0463 0416 43B9 \
  9E42 7CCC DB72 CEB9 A6B6 2283 0D24 0A80 97FC 3775 6022 008A 01CB \
  65F1 21C9 9698 7A8C 2277 7DD3 EEA4 9140"
```

Figure 3.7: Floating license with 7 licenses, bound to network card.

The license consists of the following parts:

- **Description of the license server**
  The line starting with SERVER describes the computer acting as the *license server* by stating its hostname (in the example: myservername), followed by either its host ID (00270e0ac34a) in case the license is bound to a network card or the ID of the dongle, and the number of the TCP/IP port (27000), over which the HALCON applications connect to the *license manager daemon* running on the license server.

  Please note that the **hostname must be specified correctly.** The reason is that the HALCON applications need the name to connect to the license server (even if they are started on the license server itself).

  This means that if you did not send the name of the computer acting as the license server to your distributor when requesting the license, you must adapt this entry. The same holds if you decide to switch the dongle to another computer.

  You may also need to adapt the port number, e.g., if the default port number 27000 is already used by another software in your computer network. Note, that on many systems all ports < 1024 are privileged and can only be used by privileged accounts!

- **Path to the license manager daemon**
  The line starting with VENDOR (or DAEMON) contains the path of the daemon mvtecd (see also section 4.3.3 on page 53). On Windows and Linux, this program resides in the subdirectory FLEXlm\%HALCONARCH% of the folder where you installed HALCON on the license server (see table 1.2 on page 10 for possible values of the environment variable HALCONARCH). On OS X, it resides in /Library/Application Support/HALCON-12.0/FLEXlm. In most cases, you must adapt this path. Unfortunately, you cannot use environment variables in the license file.

  If the path name of the daemon contains spaces, it will have to be enclosed in double quotes.

  Note that the daemon mvtecd opens a second port. By default, this port is selected by the operating system, and thus its number can change. Especially when using a firewall, you might need to specify the port number explicitly. For this, append the string port=*number* to the line starting with VENDOR, for example as follows (if necessary, replace x64-win64 by the value of the environment variable HALCONARCH):

```
VENDOR mvtecd "C:\Program Files\MVTec\HALCON-12.0\FLEXlm\x64-win64\mvtecd" port=28000
```

- **The license keys**
  Like the node-locked network card license depicted in figure 3.5 on page 37, the floating license contains two *license keys* (lines starting with FEATURE), one for the runtime parts of HALCON (MVTec_HALCON) and one for the development parts (MVTec_HDevelop). In contrast to the node-locked version, the floating license keys do not contain the entry HOSTID , of course, because with a floating license HALCON applications can be started on any computer in the network. Instead, the keys specify how many applications can run simultaneously in the entry after the expiration date (in the example: 7). **This part of the license file must not be modified**.

Step 5: Install the license

As in the previous sections, installing the license file means to rename the file to license.dat, if necessary, and then to place it into the subdirectory license of the folder where you installed HALCON. Unlike the possibility described in section 1.4 on page 10, where you can choose a similar name as well (e.g., license-12.0.dat), for floating licenses only license.dat is valid (but you can apply changes manually via installs.exe, see section 4.3.3.1 on page 54). Note that you must place a copy of the license file on all computers you installed HALCON on, i.e., on the license server and on all those computers you want to use HALCON on.

Step 6: (Re-)start the license manager daemon

Finally, you must start the license manager daemon, or restart it if it is already running. Please refer to section 4.3.3 on page 53 for further information.

Note that whenever you get a new floating license you must copy it to all computers and then *restart* the license manager daemon. The same is true if you modify parts of the license file while the license manager daemon is running. **Only for HALCON releases** $<$ **7.1**: If you use different HALCON versions at the same time, you have to start the different license manager daemons on different servers.

3.4 Runtime Licenses

In contrast to a development license, a runtime license only allows to run HALCON applications. Like a development license, a runtime license is permanent, but must be *node-locked* (see also section 3.1 on page 31).

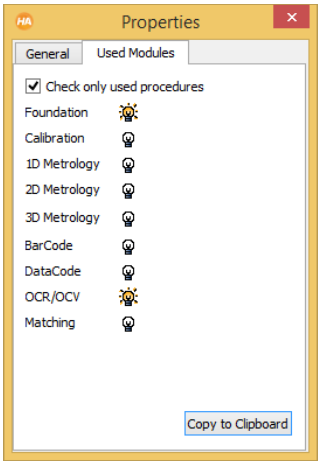
Step 1: Extract the required modules

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 67, for more information). Figure 3.8 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	○

Figure 3.8: 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 3.3.1 on page 36 (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 3.3.2 on page 37).

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, which looks similar to the one depicted in figure 3.9. If you requested a dongle-bound license, you will also receive the dongle.

If you compare the depicted license with the corresponding development license in figure 3.5 on page 37, you will note two differences: First, the runtime license contains only one *license key* for the runtime parts of HALCON (`FEATURE MVTec_HALCON`). Secondly, the entry `VENDOR_STRING` contains a different number (`1` instead of `511`). In this entry the licensed modules are stored; in the example, only 'Foundation' is licensed.

```
##########################################
# MVTec Software GmbH (ID: 00270e0ac34a) #
# Runtime Module: Foundation             #
##########################################
FEATURE MVTec_HALCON mvtecd 12.0 01-jan-0000 uncounted VENDOR_STRING=1 \
  HOSTID=00270e0ac34a ONE_TS_OK SIGN="DEAD BEEF EB14 D4A6 5557 C450 4217 \
  885A 6B02 AD22 6F1C 74DF C152 97E7 26A0 1F43 C4E2 BD29 FF44 7790 \
  2D5F 5AD1 B33C 3EF0 8DF5 DBCF 75CF D7AB 428F ACC5"
```

Figure 3.9: Runtime license for the module 'Foundation', bound to network card.

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-12.0.dat` or similar, see section 1.4 on page 10). In case of a dongle-bound please make sure that the dongle driver is installed (see section 4.3.2 on page 52).

3.5 How to Upgrade a License

Node-locked Licenses

If you upgrade a HALCON node-locked license to a newer version, e.g., from HALCON 11.0.x to HALCON 12.0, 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`. Note that the comments at the beginning of the license file reflect the history of the license, see figure 3.10 for an example upgrade from HALCON 11.0 to HALCON 12.0.

```
################################################################
# MVTec Software GmbH                                          #
# License history:                                             #
# 02-oct-2011: new license,  HALCON 11.0, ID: 00270e0ac34a     #
# 10-oct-2012: free upgrade, HALCON 12.0, ID: 00270e0ac34a     #
################################################################
FEATURE MVTec_HALCON mvtecd 12.0 01-jan-0000 uncounted VENDOR_STRING=511 \
        HOSTID=00270e0ac34a SIGN="DEAD BEEF 749D 6F3A 986E 93F9 754F EAFE 0B78 \
        B20A 9319 AFEF A7FC 9CAC B75C 049D 2ED5 F54F 3778 A8E5 6C61 4F01 \
        9C2A 84AB 1B2D 4D36 66A1 215C 6935 64E9"
FEATURE MVTec_HDevelop mvtecd 12.0 01-jan-0000 uncounted VENDOR_STRING=511 \
        HOSTID=00270e0ac34a SIGN="DEAD BEEF B31A 7CB4 0195 73D9 0463 0416 43B9 \
        9E42 7CCC DB72 CEB9 A6B6 2283 0D24 0A80 97FC 3775 6022 008A 01CB \
        65F1 21C9 9698 7A8C 2277 7DD3 EEA4 9140"
```

Figure 3.10: Upgrading a node-locked license from HALCON 11.0 to HALCON 12.0.

Floating Licenses

If you are upgrading a floating license from HALCON 7.1 or later to a newer version, e.g., HALCON 10.0, your distributor provides you with a so-called *upgrade license file*, which contains upgrade license keys (lines start with `UPGRADE`). The contents of this file have to be appended to the end of the old license file.

If you are upgrading HALCON versions older than 7.1 to a newer version, e.g., HALCON 12.0, no specific upgrade license for a floating license is available anymore. Therefore, if you want to use an older version of a floating license simultaneous to the new one, you have to start the corresponding license manager daemons on separate servers.

If the number of simultaneous users exceeds the number specified in the floating license, you can order an increment license from your distributor (lines start with `INCREMENT`). The contents of this file have to be appended to the end of the old license file.

Chapter 4

Troubleshooting

This chapter offers help for problems encountered during installing or uninstalling HALCON on Windows systems (section 4.1 and section 4.2, respectively), with the licensing mechanism (section 4.3 on page 50), or when starting HDevelop or your own HALCON applications (section 4.4 on page 58) and other miscellaneous problems.

Note that throughout the chapter the environment variable HALCONARCH is referenced. See table 1.2 on page 10 for the possible values of this variable.

4.1 Problems During Installation (Windows)

- **Registration of halconx.dll or hdevenginex.dll failed**
  On some systems you might get a warning message that the HALCON/COM interface library halconx.dll or hdevenginex.dll failed to self-register. A possible cause for this may be that the Microsoft library atl.dll was not registered properly. This library resides in the directory %SystemRoot%\system32, e.g., C:\WINNT\system32. To register the library, open a Windows Command Prompt, change into the directory %HALCONROOT%\misc\x86-win32, and execute the supplied program reg_halconx twice as follows:

  ```
  reg_halconx atl.dll
  reg_halconx
  ```

  The first call of reg_halconx registers atl.dll, the second one registers halconx.dll.

  As an alternative to the Command Prompt, you can use the dialog Start ▷ Run; here, you can select the program via the button Browse or drag and drop reg_halconx from the Windows Explorer, followed by the library name if necessary. Note that by default the Windows Explorer does not show DLL files, unless you explicitly tell it to do so.

- **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 2.8 on page 22 for information about installer switches.

4.2 Problems During Uninstallation (Windows)

- **Unregistration of** `halconx.dll`
  If you had to register the HALCON/COM interface library `halconx.dll` manually as described in section 4.1 on page 45, you must unregister it manually before you can uninstall HALCON. To do so, open a Windows Command Prompt, change into the directory `%HALCONROOT%\misc\x86-win32`, and execute the supplied program `reg_halconx` as follows:

  ```
  reg_halconx /u
  ```

  As an alternative to the Command Prompt, open the dialog `Start ▷ Run`, select the program `reg_halconx` via the button `Browse`, append the option `/u`, and then click `OK`.

- **"Internal Error" (does only occur when using HALCON versions prior to 9.0)**
  If the uninstall process terminates with a message like

  > **"Internal Error, unable to load or call external DLL. Please contact your distributor for more information."**

  the most likely reason is that a new HALCON version was installed over an existing one without completely removing the old files first.

  Other possible reasons might be that the whole HALCON directory was moved to another position on the hard disk, or the environment variable `HALCONROOT` was changed manually. You have to follow the following steps to recover from the error:

  1. Check whether the environment variable `HALCONROOT` matches the location of your HALCON installation. You can check this via the system control panel `System` (look for `Environment`) or in a Windows Command Prompt via `echo %HALCONROOT%`. If the variable contains nothing or the files are in some other location, you have to set `HALCONROOT` manually via the `System` control panel.

  2. Check whether the following files / directories are present:

     - In the directory `%HALCONROOT%\FLEXlm\%HALCONARCH%`: `HalconUninst.dll`
       Note that by default the Windows Explorer does not show DLL files, unless you explicitly tell it to do so.
       `HalconUninst.dll` is a HALCON specific DLL for the uninstallation process.

     - **Only for HALCON 8.0**
       In the directory `C:\Program Files\InstallShield Installation Information`: the directory `{096CE3F7-2FD9-4460-A270-F9F4740DB91B}` with a set of files. This directory has been created during the installation. If it has been removed, an automatic uninstallation is not possible.

- **Only for HALCON 7.1**
  In the directory `C:\Program Files\InstallShield Installation Information`: the directory `{0B2DE0B7-FD31-11D9-A19F-00E01883F42C}` with a set of files. This directory has been created during the installation. If it has been removed, an automatic uninstallation is not possible.

- **Only for HALCON 7.0**
  In the directory `C:\Program Files\InstallShield Installation Information`: the directory `{47F424B4-1077-11D8-A0D3-00E01883F42C}` with a set of files. This directory has been created during the installation. If it has been removed, an automatic uninstallation is not possible.

- **Only for HALCON 6.1**
  In the directory `C:\Program Files\InstallShield Installation Information`: the directory `{BBEC9F40-4A36-11D6-A14C-00E0296C2846}` with a set of files. This directory has been created during the installation. If it has been removed, an automatic uninstallation is not possible.

- **Only for HALCON releases < 6.1**
  In your `%HALCONROOT%`directory: `Uninst.isu`
  This file has been created during the installation. If it has been removed, an automatic uninstallation is not possible.

3. Check whether the registry entry for the uninstaller is set properly: Start `regedit.exe` in a Windows Command Prompt or in the dialog `Start ▷ Run`.

   - **Only for HALCON 9.0 or higher**
     Please look for `uninstall.exe` under `%HALCONROOT%/misc/x86-win32`.

   - **Only for HALCON 8.0**
     Go to

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

     There, you should find a key named `UninstallString`. The value of this string should be something like this:

     ```
     RunDll32 C:\PROGRA~1\COMMON~1\INSTAL~1\engine\6\INTEL3~1\Ctor.dll,
     LaunchSetup "C:\Program Files\InstallShield Installation
     Information\{096CE3F7-2FD9-4460-A270-F9F4740DB91B}
     setup.exe" -l0x9 UninstallHALCON
     ```

   - **Only for HALCON 7.1**
     Go to

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

There, you should find a key named `UninstallString`. The value of this string should be something like this:

```
RunDll32 C:\PROGRA~1\COMMON~1\INSTAL~1\engine\6\INTEL3~1\Ctor.dll,
LaunchSetup "C:\Program Files\InstallShield Installation
Information\{0B2DE0B7-FD31-11D9-A19F-00E01883F42C}\
setup.exe" -l0x9 UninstallHALCON
```

- **Only for HALCON 7.0**
  Go to

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

  There, you should find a key named `UninstallString`. The value of this string should be something like this:

  ```
  RunDll32 C:\PROGRA~1\COMMON~1\INSTAL~1\engine\6\INTEL3~1\Ctor.dll,
  LaunchSetup "C:\Program Files\InstallShield Installation
  Information\{47F424B4-1077-11D8-A0D3-00E01883F42C}\
  setup.exe" -l0x9 UninstallHALCON
  ```

- **Only for HALCON 6.1**
  Go to

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

  There, you should find a key named `UninstallString`. The value of this string should be something like this:

  ```
  RunDll32 C:\PROGRA~1\COMMON~1\INSTAL~1\engine\6\INTEL3~1\Ctor.dll,
  LaunchSetup "C:\Program Files\InstallShield Installation
  Information\{BBEC9F40-4A36-11D6-A14C-00E0296C2846}\
  setup.exe" -l0x9 UninstallHALCON
  ```

- **Only for HALCON releases $<$ 6.1**
  Go to

  ```
  HKEY_LOCAL_MACHINE -> SOFTWARE -> Microsoft -> Windows ->
    CurrentVersion -> Uninstall -> HALCON <Version-Number>
  ```

  with `<Version-Number>` being the version number of the HALCON version that you want to uninstall. There, you should find a key named `UninstallString`. The value of this string should be something like this (the option `-c...` is only included if you have chosen an installation with floating licenses):

```
C:\WINNT\IsUninst.exe
-f"C:\Program Files\MVTec\HALCON\Uninst.isu"
-c"C:\Program Files\MVTec\HALCON\FLEXlm\i586-nt4\
HalconUninst.dll"
```

Make sure that the path `C:\Program Files\MVTec\HALCON` (or equivalent) in the above example points to the folder where you installed HALCON. Note that **file names that contain blanks must be quoted** as in the above example. If you encounter unquoted path names containing blanks, please insert the quotation marks yourself.

4. Close the registry editor and try to run the uninstaller again.

- **Uninstallation failed**
  If the automatic uninstallation fails for another reason, proceed as follows:

  - For floating licenses only: Uninstall the license manager daemon as described in section 4.3.3.1 on page 54.

  - 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.

4.3 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 information identifying your network card or dongle matches the entries in the license file (see the corresponding sections in chapter 3 on page 31). If the two do not match, please send the new identifying information to your distributor. See section 4.3.1 if you encounter problems with extracting the identifying information.

Section 4.3.2 on page 52 contains information all around dongle drivers, e.g., how to check whether they are installed correctly.

If you have problems with a floating license, the first step is to check whether the entries in the license file that can be customized contain the correct information, especially the port number for the license manager daemon (see section 3.3.3 on page 38). If the license file is correct, please refer to section 4.3.3 on page 53, which explains how to check whether the floating license manager daemon was installed successfully and how to install it manually.

The FlexNet Publisher End User's Guide can be obtained from `http://www.flexera.com`.

4.3.1 Extracting Host IDs

- **The dialog `Help ▷ About HDevelop` does not show any host IDs**

  If HDevelop fails to detect any host IDs although your computer does have a network board or a dongle, please try to extract the host IDs manually using the program `lmhostid` shipped together with the license manager FLEXlm. Under Windows, open a Windows Command Prompt[1]. Under Linux, open a shell, change into the directory `$HALCONROOT/FLEXlm/$HALCONARCH`.

  - To identify a computer by its **network card**, type `lmhostid -ether` (or just `lmhostid`). The output might look like this on a Windows system:

    ```
    > lmhostid -ether
    lmhostid - Copyright (c) 1989-2014 Flexera Software LLC. All Rights Reserved.
    The FlexNet host ID of this machine is "00270e0ac34a"
    ```

    If `lmhostid` returns `"ffffffff"` or `"0"` please see below.

  - To check the **dongle ID**, type `lmhostid -flexid`. The output now might look like this:

    ```
    > lmhostid -flexid
    lmhostid - Copyright (c) 1989-2014 Flexera Software LLC. All Rights Reserved.
    The FlexNet host ID of this machine is "FLEXID=9-1135eb58"
    ```

    You can safely ignore the warning 'WIBU DLL not found'. The host ID must be identical to the one printed on the back of the dongle. If this is not the case, please see below.

- **`lmhostid -ether` returns `"ffffffff"` or `"0"`**

  If `lmhostid` returns `"ffffffff"` or `"0"`, this usually indicates that you do not have a network card. Check whether the client for Microsoft networks is installed in the dialog `Start` ▷ `Settings` ▷ `Network Connections` ▷ `Local Area Connections` ▷ `Properties`. If the client for Microsoft networks does not appear in the list install it by clicking `Install` and then selecting it from the list of clients.

  Also make sure the `Workstation` service is started. See section 4.3.3.1 on page 54 for how to check if a service is running.

  Then, call `lmhostid` again or open HDevelop's dialog `Help` ▷ `About` (see section 3.1 on page 31) to check whether a correct host ID is found now.

- **Multiple IDs for only one network card**

  Sometimes more than one ID is returned even if there is only a single network card. In this case, use the ID that remains when both the NetBEUI and IPX/SPX protocols are disabled. Alternatively, use the ID that does not change when you reboot your computer.

[1] Do not start the program from the Windows Explorer. You *must* use a Command Prompt.

- **Licensing via network card does not work when network is not connected**

  You must disable Media Sense (*DHCP*) when no network is connected to the network card. Usually, it is sufficient to disable the DHCP protocol in this case. Further details can be found under `http://support.microsoft.com/support/kb/articles/Q239/9/24.asp`.

- `lmhostid -flexid` **does not return the dongle ID**

  If `lmhostid` does not return the ID that is printed on the back of the dongle, check whether the dongle driver is installed and configured correctly as described in section 4.3.2.

  If the driver is installed but `lmhostid` still does not return the correct ID, please check the corresponding USB port of your computer, before requesting a new dongle.

4.3.2 Dongle Drivers

4.3.2.1 Dongles for the USB Port (Windows)

As described in section 2.2.1 on page 14, you can let HALCON install the driver programs necessary for using dongles. You can check the success of this installation in the following system dialog, which should contain two entries called `hardlock` and `haspnt`.

- `Start` ▷ `Programs` ▷ `Accessories` ▷ `System Tools` ▷ `System Information` ▷ `Software Environment`, then select `Drivers` (or `System Drivers`).

Depending on your operating system, you can install, configure, and uninstall the dongle drivers manually using an auxiliary program that is part of each HALCON installation. Open a Windows Command Prompt or the dialog `Start ▷ Run` and execute the program `%HALCONROOT%\FLEXlm\%HALCONARCH%\flexid9\haspdinst.exe` in one of the following ways:

```
haspdinst                opens a dialog explaining how to use haspdinst
haspdinst -info          shows the status of the drivers
haspdinst -i             installs the drivers
haspdinst -remove        removes the drivers
```

Please note that you need administrator privileges to install the drivers!

Note that in order to **execute command line tools with administrator privileges**, you will need to open a command shell using “Run as Administrator” and execute the tool from there (even if you are already logged in as administrator).

After installing the driver, you need to copy the architecture-dependent third-party FlexNetID9 library from `FLEXlm\%HALCONARCH%\flexid9` to the Windows system directory. The DLL is named `haspsrm_win64.dll` for 64-bit systems, and must be copied to `%WINDIR%\System32`. On 32-bit systems or 64-bit systems using the 32-bit version of HALCON, the DLL `haspsrm_win32.dll` must be copied to `%WINDIR%\System32` or `%WINDIR%\SysWOW64`, respectively.

Note that the HALCON uninstallation process does not uninstall the driver as it might also be used by another application. However, the uninstallation removes the auxiliary program. Therefore, if you want to uninstall the drivers, do so before uninstalling HALCON.

See `%HALCONROOT%\FLEXlm\%HALCONARCH%\flexid9\readme.html` for further information.

4.3.2.2 Dongles for the USB Port (Linux)

In order to install the USB dongle daemon on Linux, issue the following commands as root:

```
# cd $HALCONROOT/FLEXlm/$HALCONARCH/flexid9
# cp libhasp_linux_x86_64.so /usr/lib
# ./dinst .
```

The installation script will install the dongle daemon and the corresponding start-up scripts.

```
---------------------------------------
Copy AKSUSB daemon to /usr/sbin ...
Copy WINEHASP daemon to /usr/sbin ...
Copy HASPLMD daemon to /usr/sbin ...
Copy start-up script to /etc/init.d ...
Link HASP SRM runtime environment startup script to system startup folder
Starting HASP SRM runtime environment ...
Starting AKSUSB daemon:                                    [  OK  ]
Starting WINEHASP daemon:                                  [  OK  ]
Starting HASPLM daemon:                                    [  OK  ]
Done
---------------------------------------
```

To test, if the dongle daemon is running, enter the following command as root (where /etc/init.d is to be replaced by the directory reported when installing the dongle daemon):

```
# /etc/init.d/aksusbd status
AKSUSB daemon is running!
WINEHASP daemon is running!
HASPLM daemon is running!
```

To uninstall the dongle daemon, enter the following command as root:

```
# ./dunst
Stopping HASP SRM RTE ...
Stopping HASPLM daemon:                                    [  OK  ]
Stopping WINEHASP daemon:                                  [  OK  ]
Stopping AKSUSB daemon:                                    [  OK  ]
Removing files ...
Done
```

See $HALCONROOT/FLEXlm/$HALCONARCH/flexid9/readme.html for further information.

4.3.3 The License Manager Daemon

The license manager daemon (which is used in case of floating licenses) consists of two programs:

1. `lmgrd`: This is the main license manager daemon, which is provided by the licensing software FLEXlm. It handles the connections from the HALCON applications and passes them on to the second daemon.

2. `mvtecd`: This is the so-called *vendor daemon*, which is charged with keeping track of the HALCON applications currently being run.

The following sections explain how to install, (re-)start, and uninstall the license manager daemon on Windows (section 4.3.3.1), Linux (section 4.3.3.2 on page 56), and OS X (section 4.3.3.3 on page 57) platforms. **Only for HALCON releases < 7.1**: Note that if you use different HALCON versions simultaneously you have to start the corresponding license manager daemons on separate servers.

4.3.3.1 Windows

How to Install the License Manager Daemon

If you did not install the license manager daemon via the setup as described in section 2.2.1 on page 14, or if its installation failed, you can install it manually at a later time as described below. Note that **you need administrator privileges** for this procedure.

Note that in order to **execute command line tools with administrator privileges**, you will need to open a command shell using “Run as Administrator” and execute the tool from there (even if you are already logged in as administrator).

The license server itself does not require a full installation of HALCON. In case you do not want to install HALCON on the license server at all, the required files are installed in the following way: Using the downloaded installer, you can install the required files by selecting the option “Compact” (make sure that “Install floating license server” is also selected).

Open a Windows Command Prompt, change into the directory `%HALCONROOT%\FLEXlm\%HALCONARCH%`, and type (*one* long command line):

```
installs -n "HALCON Licenses"
  -c "%HALCONROOT%\license\license.dat"
  -l "%HALCONROOT%\license\license.log"
  -e "%HALCONROOT%\FLEXlm\%HALCONARCH%\lmgrd.exe"
```

The environment variable `HALCONROOT` points to the folder where you installed HALCON (see also section A.2 on page 65). The quotes are necessary to handle paths that contain blanks. As mentioned in section 3.3.3 on page 38, here you can change the name of the license file manually (replace `license.dat`, e.g., by `license-12.0.dat` or similar).

To check whether the installation succeeded, (re-)open the Windows dialog showing the state of the installed services (see above), which now should contain the entry `HALCON Licenses` (or, to be more exact, an entry with the name you specified with the option `-n` or `FLEXlm license manager` if you leave out this option).

Note that if you installed the license manager daemon manually, you must also uninstall it manually before you uninstall HALCON. Otherwise it remains installed.

How to Check Whether the License Manager Daemon is Running

To check whether the daemon was installed and started successfully:

- Open the Windows task manager using Ctrl+Shift+Esc.
- Activate the tab `Services`.

  If the daemon was started successfully by the setup program, the dialog contains the entry `HALCON Licenses`.
- Click the button `Services...` to control the state of the daemon.

How to Start the License Manager Daemon

After the installation, the daemon is not started automatically. You can start it by rebooting the computer. If you have administrator privileges, you can start it directly in the dialog which shows the state of the installed services (see above) by selecting it and then clicking the start button or icon; if this fails, please check whether the entries in the license file are correct.

How to Restart the License Manager Daemon

Whenever you replace a floating license or modify it you must *restart* the license manager daemon. Another reason for restarting the license manager daemon is when you update to a HALCON release that uses a newer version of the FLEXlm license manager software. One possible method is to reboot the computer acting as the license server. If you have administrator privileges, you can also restart the daemon manually: Open the dialog showing the state of the installed services (see above), select the entry `HALCON Licenses` and then stop and start it again.

How to Uninstall the License Manager Daemon

Please note that for the following procedure **you need administrator privileges**.

Before you can uninstall the license manager daemon, you must first stop it: Open the Windows dialog showing the state of the installed services (see above), select the entry `HALCON Licenses` and then click the stop button or icon. Note that under Windows Vista and Windows 7 an error message may appear, which can safely be ignored.

Now, open a Windows Command Prompt, change into the directory `%HALCONROOT%\FLEXlm\%HALCONARCH%`, and type:

```
installs -r -n "HALCON Licenses"
```

To check whether the uninstallation succeeded, re-open the services dialog, which should not contain an entry `HALCON Licenses` anymore.

How to Avoid License Errors with Floating Dongle Licenses

A race condition might occur if a HALCON application starts up immediately after booting the license server: The license manager might be running before the dongle driver is loaded, resulting in a license error. To avoid this problem, it is sufficient to define dependencies for the services started on the license server at boot time.

To do this, you need to edit the registry. Run `regedt32`, and navigate to the following path:

```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HALCON Licenses
```

Add the value `DependOnService` (of type `REG_MULTI_SZ`) if it does not exist already. Double-click the value and enter the following dependencies (one entry per line):

```
akshasp
aksusb
Hardlock
Haspnt
```

4.3.3.2 Linux

How to Install the License Manager Daemon

On Linux systems, the license manager daemons `lmgrd` and `mvtecd` are automatically "installed" in the subdirectory `FLEXlm/$HALCONARCH` of the directory you installed HALCON in.

How to Start the License Manager Daemon

The license manager daemon `lmgrd` must be started from the appropriate startup file (called, e.g., `/etc/init.d/boot.local`, `/sbin/init.d/boot.local`, `/etc/rc.boot`, `/etc/rc.local`, or `/etc/localrc`, please consult your system's documentation). Add the following (long) line to this file, replacing the environment variables with their content (see section A.2 on page 65):

```
   $HALCONROOT/FLEXlm/$HALCONARCH/lmgrd -c $HALCONROOT/license/license.dat
       > $HALCONROOT/license/license.log 2>&1 &
```

!

If you copied `lmgrd` to another location, you must of course adapt the path to it accordingly. Please note, that you **should not start `lmgrd` with root privileges**!

`lmgrd` automatically starts the vendor daemon `mvtecd`.

How to Restart the License Manager Daemon

Whenever you replace a floating license or modify it you must restart the license manager daemon. Another reason for restarting the license manager daemon is when you update to a HALCON release that uses a newer version of the FLEXlm license manager software.

One possible method is to reboot the computer acting as the license server. If you have administrator privileges, you can also restart the daemon manually by stopping the process `lmgrd` and then restarting it with the line used in the startup file (see above).

How to Uninstall the License Manager Daemon

To uninstall the license manager daemon you must remove the corresponding entry in the startup file (see above) and stop the currently running lmgrd and mvtecd processes. This can be achieved by rebooting the computer that acts as the license server or by use of the kill command (please see your system's documentation or ask your system administrator for advice).

4.3.3.3 OS X

How to Install the License Manager Daemon

On OS X systems, the license manager daemons lmgrd and mvtecd are automatically installed in the directory /Library/Application Support/HALCON-12.0/FLEXlm.

How to Start the License Manager Daemon

Background processes are started via launchd. The required property list file is installed in /Library/Application Support/HALCON-12.0/FLEXlm/com.mvtec.halcon.flexlm.plist:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
                       "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Label</key>
<string>com.mvtec.halcon.flexlm</string>
<key>RunAtLoad</key>
<true/>
<key>UserName</key>
<string>nobody</string>
<key>ProgramArguments</key>
<array>
<string>/Library/Application Support/HALCON-12.0/FLEXlm/lmgrd</string>
<string>-z</string>
<string>-c</string>
<string>/Library/Application Support/HALCON-12.0/license/license.dat</string>
</array>
<key>ServiceDescription</key>
<string>HALCON license manager daemon</string>
</dict>
</plist>
```

Copy the property list file to /Library/LaunchDaemons. The file needs to belong to root, and must not be writeable by anyone but the owner, or launchd will refuse to load it.

If your license file is not named license.dat, you must adapt the property list file accordingly.

The license manager will start the next time you boot the system. To start it immediately without rebooting, execute the command

```
sudo launchctl load /Library/LaunchDaemons/com.mvtec.halcon.flexlm.plist
```

To troubleshoot problems with the license, you can tell `launchd` to redirect the license manager's standard and error output to a file by adding the `StandardOutPath` and `StandardErrorPath` keys to the property list file. Keep in mind that the license manager is run as the user 'nobody', so the path you choose for the output file must be writeable by that user.

Please see the `launchctl` and `launchd.plist` man pages for more details.

How to Restart the License Manager Daemon

Whenever you replace a floating license or modify it, you must restart the license manager daemon. Another reason for restarting the license manager daemon is when you update to a HALCON release that uses a newer version of the FLEXlm license manager software.

One possible method is to reboot the computer acting as the license server. If you have administrator privileges, you can also restart the daemon manually by using the following commands:

```
sudo launchctl unload /Library/LaunchDaemons/com.mvtec.halcon.flexlm.plist
sudo launchctl load /Library/LaunchDaemons/com.mvtec.halcon.flexlm.plist
```

How to Uninstall the License Manager Daemon

To uninstall the license manager daemon, you must remove the file `/Library/LaunchDaemons/com.mvtec.halcon.flexlm.plist`.

4.4 Troubleshooting in HDevelop or HALCON Applications

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.
  - In case of floating licenses: There are too many applications using HALCON active, i.e., the maximum number of simultaneous HALCON applications (which is specified in the floating license) has been exceeded.
  - In case of node-locked licenses: If there is more than one user trying to use HALCON via remote access, the second user gets an error message.

- **Lost connection to license server**

  Verify that the license server is running. You may also check whether your machine is properly connected with the server. For this you may need to contact your system administrator.

- **No license for this operator**

  The operator which you try to execute belongs to a HALCON module that is not licensed (see section 3.4 on page 41. 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 65).

- **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 ≥ 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 65),
- the registry keys on Windows systems (section A.3 on page 67), and

A.1 The Installed File Structure

Let's take a look at the installed file structure in the directory `%HALCONROOT%` on Windows/Linux. On OS X, most of these files are installed in `/Library/Frameworks/HALCON.framework` 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.

- **`FLEXlm`:** This directory contains programs used for licensing in subdirectories corresponding to the different platforms (see chapter 3 on page 31 and section 4.3.3 on page 53) including dongle installers.

  On OS X, this directory resides in `/Library/Application Support/HALCON-12.0/FLEXlm`.

- **`bin`:** This directory contains HALCON programs, for example HDevelop (Windows: `hdevelop.exe`; Linux: `hdevelop`), again in subdirectories corresponding to the different platforms. For Windows, this directory also contains the DLLs of the HALCON libraries, the DLLs for the supported image acquisition interfaces, and IO device interfaces. The subdirectories `dotnet20`, and `dotnet35` contain the HALCON/.NET assemblies based on .NET Framework 2.0, and .NET Framework 3.5, respectively.

  On OS X, the corresponding files reside in `/Applications`.

- **`calib`:** This directory contains description files for the calibration plates, which you can use to calibrate your camera.

- doc\html: Here, you find the HTML documentation of the HALCON Image Acquisition Interfaces (subdirectory reference\acquisition), I/O device interfaces (subdirectory reference\io), some of the User's Manuals (subdirectory manuals), and the Reference Manual (subdirectory reference\operators) including the API reference of the HALCON Codelets (subdirectory reference\codelets).

  On OS X, the files reside in /Library/Application Support/HALCON-12.0/doc/html.

- doc\index: Here, you find the index files that are relevant for the full-text search of documentation.

  On OS X, the files reside in /Library/Application Support/HALCON-12.0/doc/index.

- doc\macros: Here, you find the style files for the HALCON reference manuals in all translated languages. These files set the layout and provide macros for names and formatting.

  On OS X, the files reside in /Library/Application Support/HALCON-12.0/doc/macros.

- doc\pdf: Here, you find the PDF version of the User's Manuals (subdirectory manuals), the Solution Guide I, the Solution Guide II, and Solution Guide III (subdirectory solution_guide), and of the Reference Manual in HDevelop syntax (subdirectory reference).

  On OS X, the files reside in /Library/Application Support/HALCON-12.0/doc/pdf.

- filter: This directory contains predefined filter masks.

- genicam: This directory contains the underlying GenAPI runtime software for the GigEVision, GenICamTL, and USB3 Vision 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. Those 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: This directory contains the header files that are necessary to use HALCON within the programming languages C or C++.

  On OS X, these files reside in /Library/Frameworks/HALCON.framework/Headers.

- lib: This directory contains the HALCON libraries and under Linux also the libraries for the supported image acquisition interfaces (Windows: file extension .lib; Linux: file extension .so), again in subdirectories corresponding to the different platforms.

  On OS X, the corresponding directories reside in subdirectories of /Library/Frameworks, e.g., in /Library/Frameworks/HALCONCpp.framework for C++.

- license: The license file must be placed here (see section 3.1 on page 31).

  On OS X, this directory resides in /Library/Application Support/HALCON-12.0/license.

- lut: This directory contains predefined look-up tables.

- misc: This directory contains miscellaneous files for the installer, e.g., the GigE Vision filter driver.

- ocr: This directory contains pretrained fonts.

- procedures: This directory contains the external procedures for HDevelop and HDevEngine.

The subdirectories of the directory %HALCONEXAMPLES% contain example programs for the different parts of the HALCON system. If %HALCONEXAMPLES% is not set, %HALCONROOT%\examples will be used as a fallback.

On OS X, these files reside in /Users/Shared/Library/Application Support/HALCON-12.0/examples.

- 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 23 on page 195).
- c#: Examples for using HALCON within the programming language C# (see also the Programmer's Guide, section 17.1 on page 149).
- codelets: HALCON Codelets (see the Programmer's Guide, section 17.5 on page 156, for more information).
- cpp: Examples for using HALCON within the programming language C++ (see also the Programmer's Guide, chapter 13 on page 115).
- cpp.net: Examples for using HALCON within managed C++ (see also the Programmer's Guide, section 17.1 on page 149).
- delphi: Examples for using HALCON within Borland Delphi.
- 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 Reference Manual 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 17.1 on page 149).

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 HAL-

CON/.NET assembly with a local path, thus after copying you must restore the reference (see the Programmer's Guide, page 129). You can also modify the environment variable HALCONEXAMPLES to point to the new location.

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 17, 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 61.

  - `%HALCONROOT%\help`
    The files in this directory act as the HALCON information database (see section A.1 on page 61 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 3 on page 31).

  - %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 9.

- 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 2.13 on page 27 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 25.

A.2.4 General Environment Variables

- PATH
  Windows: During the installation, the directories %HALCONROOT%\bin\%HALCONARCH% and %HALCONROOT%\FLEXlm\%HALCONARCH% are 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 vari-

able 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, e.g., 12.0.

Please note: On x64-based systems HKLM\SOFTWARE\MVTec has to be replaced by 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\12.0\Initialized\InstallDir
```

```
HKLM\SOFTWARE\MVTec\HALCON\12.0\Initialized\ProgramGroup
HKLM\SOFTWARE\MVTec\HALCON\12.0\Initialized\Architecture
HKLM\SOFTWARE\MVTec\HALCON\12.0\Initialized\ExampleRoot
HKLM\SOFTWARE\MVTec\HALCON\12.0\Initialized\LanguageDoc
HKLM\SOFTWARE\MVTec\HALCON\12.0\Initialized\InstalledVersion
HKLM\SOFTWARE\MVTec\HALCON\12.0\Initialized\InstalledRevision
HKLM\SOFTWARE\MVTec\HALCON\12.0\Initialized\InstalledBuild
```

Sections selected during installation:

```
HKLM\SOFTWARE\MVTec\HALCON\12.0\InstalledComponents
```

(depending on the sections)

A.3.4 Uninstall Information

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
InstallLocation
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
UninstallString
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
DisplayIcon
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
DisplayName
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
DisplayVersion
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
Publisher
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
Version
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
URLInfoAbout
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
URLUpdateInfo
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
NoModify
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MVTec HALCON 12.0\
NoRepair
```

A.3.5 HalconX

```
HKCR\HalconX.HOperatorSetX
HKCR\HalconX.HFeatureSetX
HKCR\HalconX.HTemplateX
```

```
HKCR\HalconX.HNCCModelX
HKCR\HalconX.HShapeModelX
HKCR\HalconX.HObjectModel3DX
HKCR\HalconX.HSurfaceModelX
HKCR\HalconX.HSurfaceMatchingResultX
HKCR\HalconX.HShapeModel3DX
HKCR\HalconX.HDeformableModelX
HKCR\HalconX.HDescriptorModelX
HKCR\HalconX.HVariationModelX
HKCR\HalconX.HCalibDataX
HKCR\HalconX.HCameraSetupModelX
HKCR\HalconX.HStereoModelX
HKCR\HalconX.HComponentTrainingX
HKCR\HalconX.HComponentModelX
HKCR\HalconX.HGnuplotX
HKCR\HalconX.HSocketX
HKCR\HalconX.HSerialX
HKCR\HalconX.HMeasureX
HKCR\HalconX.HMatrixX
HKCR\HalconX.HFunction1DX
HKCR\HalconX.HBarCode1DX
HKCR\HalconX.HBarCode2DX
HKCR\HalconX.HBarCodeX
HKCR\HalconX.HDataCode2DX
HKCR\HalconX.HMiscX
HKCR\HalconX.HImageX
HKCR\HalconX.HRegionX
HKCR\HalconX.HUntypedObjectX
HKCR\HalconX.HXLDX
HKCR\HalconX.HXLDContX
HKCR\HalconX.HXLDPolyX
HKCR\HalconX.HXLDParaX
HKCR\HalconX.HXLDModParaX
HKCR\HalconX.HXLDExtParaX
HKCR\HalconX.HWindowX
HKCR\HalconX.HTupleX
HKCR\HalconX.HFileX
HKCR\HalconX.HFramegrabberX
HKCR\HalconX.HOCRBoxX
HKCR\HalconX.HOCRMlpX
HKCR\HalconX.HOCRSvmX
HKCR\HalconX.HLexiconX
HKCR\HalconX.HOCVX
HKCR\HalconX.HClassBoxX
HKCR\HalconX.HClassMlpX
HKCR\HalconX.HClassSvmX
HKCR\HalconX.HClassGmmX
```

```
HKCR\HalconX.HColorTransLUTX
HKCR\HalconX.HBgEstiX
HKCR\HalconX.HSystemX
HKCR\HalconX.HInfoX
HKCR\HalconX.HPoseX
HKCR\HalconX.HHomMat2DX
HKCR\HalconX.HHomMat3DX
HKCR\HalconX.HQuaternionX
HKCR\HalconX.HSheetOfLightModelX
HKCR\HalconX.HClassLUTX
HKCR\HalconX.HComputeDeviceX
HKCR\HalconX.HMutexX
HKCR\HalconX.HEventX
HKCR\HalconX.HConditionX
HKCR\HalconX.HBarrierX
HKCR\HalconX.HTextModelX
HKCR\HalconX.HTextResultX
HKCR\HalconX.HMetrologyModelX
HKCR\HalconX.HSerializedItemX
HKCR\HalconX.HClassKnnX
HKCR\HalconX.HOCRKnnX
HKCR\HalconX.HClassTrainDataX
HKCR\HalconX.HScatteredDataInterpolatorX
HKCR\HalconX.HSampleIdentifierX
HKCR\HalconX.XLDDistTransX
HKCR\HalconX.DrawingObjectX
HKCR\HalconX.Scene3DX
HKCR\HalconX.BeadInspectionModelX
HKCR\HalconX.HWindowXCtrl
HKCR\HalconX.HDevWindowStackX
```

A.3.6 HDevEngineXLib

```
HKCR\HDevEngineXLib.HDevEngineX
HKCR\HDevEngineXLib.HDevProgramX
HKCR\HDevEngineXLib.HDevProgramCallX
HKCR\HDevEngineXLib.HDevProcedureX
HKCR\HDevEngineXLib.HDevProcedureCallX
HKCR\HDevEngineXLib.HDevOperatorImplX
```

Index