

# HDevRunner - Runs your vision script

## Manual

Documentation for the HDevRunner Standard Edition and HDevRunner Professional Edition 1.0.2

Copyright 2017 Heindl Solutions

For further information go to <http://www.hdevrunner.com/>



- Microsoft, Visual Studio, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and other countries.
- HALCON and HDevelop are trademarks of MVTec Software GmbH, Germany.
- Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

# HDevRunner Manual

- [Preface](#)
  - [Introduction](#)
  - [Prerequisites](#)
  - [Bug reports](#)
  - [Installation](#)
  - [Examples](#)
- [Introduction](#)
  - [What is HDevRunner?](#)
  - [Why use HDevRunner?](#)
  - [A first example: Hello World](#)
- [Features of HDevRunner Standard Edition](#)
  - [Quickstart](#)
  - [Run with command line arguments](#)
- [Features of HDevRunner Professional Edition](#)
  - [Quickstart](#)

# Preface

## Introduction

HDevRunner is the perfect combination for [MVTec HALCON](#) to deploy your algorithms to customer machines - fast and easy. No more C++ or C# development needed. In addition, the Professional Edition runs your scripts on multiple images during development, resulting in an unprecedented speed in evaluating your algorithm.

## Prerequisites

HDevRunner is a tool to be used with [HALCON](#), the machine vision library from MVTec Software GmbH. It does not make sense to use it without an active HALCON installation on the same machine, it wouldn't even be able to start. Depending on what your tasks are, the runtime license of HALCON might be enough to use HDevRunner.

## Bug reports

Although every attempt has been made to make HDevRunner bug-free, one cannot rule out that there will still be bugs left in the product. Please report bugs to [bugs@hdevrunner.com](mailto:bugs@hdevrunner.com).

## Installation

HDevRunner can be downloaded for free at <http://www.hdevrunner.com/downloads>. It runs as HDevRunner Demo version by default, but it can be upgraded to HDevRunner Standard or HDevRunner Professional by purchasing a license. A license can be purchased from Heindl Solutions at <http://www.hdevrunner.com/buy>.

At the moment, HDevRunner Windows 7, Windows 8, Windows 8.1, and Windows 10 are supported. Linux support is planned to be released in Q1/2018.

Save the downloaded file (e.g. `install-hdevrunner-1.0.0.exe`) in a temporary directory and double-click the installer file. You will have to accept the Windows dialog > Do you want to allow this app from an unknown publisher to make changes to your device?

During installation, you have to accept the License Agreement in order to use the product. The x86 version of HDevRunner is a good default, thus selecting the components *Program Files (x86)* and *Desktop Shortcuts (x86)* makes sense. Of course you must have the corresponding *x86sse2-win32* architecture for HALCON installed. If you have a *x64-win64* architecture of HALCON installed, you can select to install the components *Program Files (x64)* and *Desktop Shortcuts (x64-win64)* instead.

Select the *License Center for HDevRunner* component to simplify licensing of HDevRunner. You can skip this components and call HDevRunner with the corresponding command line arguments. See section 'Licensing' for details.

The destination folder is `C:\Program Files\Heindl Solutions\HDevRunner` by default. We recommend to keep this default. Changing it might lead to problems with some paths in the provided example scripts.

HDevRunner depends on the *Microsoft Visual Studio 14.0 Runtime*. This will be installed automatically during the installation.

## Examples

The Examples directory, typically installed at `*C:\Filesolutions` contains a variety of examples of using HDevRunner.

# Introduction

## What is HDevRunner?

HDevRunner is a tool that allows you to execute HDevelop scripts, external procedures, and library procedures without using HDevelop and without the need for a HALCON development license.

HDevRunner is available in three different editions.

- HDevRunner Demo is ideal to try out most of the functionality of the HDevRunner Standard and HDevRunner Professional editions for free. It must not be used for commercial purposes.
- HDevRunner Standard is ideal if you need a fast and easy way to create a GUI application running your vision algorithms.
- HDevRunner Professional contains all the advantages of the Standard edition, but includes many additional features which make it ideal for the advanced professional HALCON developer.

## Why use HDevRunner?

HDevRunner makes it very fast and simple to run your HALCON vision scripts. That's the main task for the HDevRunner Standard Edition. The HDevRunner Professional Edition provides additional features for developers and also aims at improving your development experience with HDevelop. For example, the watch mode allows you to watch procedure directories and arbitrary additional directories for file changes. If a script or e.g. test images get modified, the script will be reloaded and executed immediately. This is very useful to re-run your scripts for testing directly after saving a script / pressing CTRL+S in HDevelop.

As a experienced HALCON developer, you might choose to write a hosting GUI application that integrates HDevEngine and your HALCON scripting code. That way, you can create arbitrarily complex GUIs in a development environment of your choice. For full-featured applications, this will be a valid approach. But it is often the case that you want to execute additional scripts on the customer machine, e.g. for (re-)calibration or testing a new camera. Instead of rewriting big parts of your main application logic, you can then use HDevRunner to execute the scripting code. Also, it is not a simple task to get the HDevEngine integrated in your host application such that HALCON behaves exactly like in the HDevelop development environment. Frustrating trial and error sessions and/or major rewritings of your scripts will follow.

## A first example: Hello World

A simple hello-world.hdev HDevelop script might look like this:

```
open_file ('standard', 'output', FileHandle)
fwrite_string (FileHandle, 'Hello World!\n')
close_file (FileHandle)
```

Create this scripting code or open the example `examples/hello-world/hello-world.hdev`.

Now open the HDevRunner Command Prompt from the Windows Start Menu. Type or copy and paste:

```
hdevrunner "C:\Program Files\Heindl Solutions\HDevRunner\examples\hello-world\hello-world.hdev"
```

HDevRunner will be started, the script `hello-world.hdev` will be loaded, and the script will be executed.

The output will be similar to the following:

```
HDevRunner 1.0.0 Professional Edition - Copyright 2017 Heindl Solutions
Hello World!
```

## Modifying the Hello World example

As a simple exercise you might want to switch to Watch Mode using `--watch` in the previous command like so:

```
hdevrunner --watch --no-default-window "C:\Program Files\Heindl Solutions\HDevRunner\examples\hello-world\hello-world.hdev"
```

HDevRunner will be started, the script `hello-world.hdev` will be loaded, the script will be executed and HDevRunner will have a watch for changes on the directory `C:\Program Files\Heindl Solutions\HDevRunner\examples\hello-world`. Especially, HDevRunner will not exit automatically. You can exit HDevRunner by focusing the HDevRunner Console and pressing ESC. The command line argument `--no-default-window` is necessary because HDevRunner normally opens a default Graphics Window on startup (similar to HDevelop) but our simple example doesn't output any graphics and we want to avoid an empty default Graphics Window to be shown.

Now try to change you script code, e.g. add these lines to your HDevelop script `hello-world.hdev`:

```
get_system ('version', Version)
fwrite_string (FileHandle, 'You are using HALCON ' + Version + '\n')
```

before the `close_file` operator. Each time you save the script during writing of the code, the script gets executed.

Example output during some development steps might be the following. Please note how the error in calling `get_system` with the (wrong) query `halconversion` is immediately visible:

```
HDevRunner 1.0.0 Professional Edition - Copyright 2017 Heindl Solutions
Watching for directory changes or key presses in the console window... Press 'ESC' to exit.
Hello World!
Execution of program P:\projects\hdevrunner\hdevrunner\examples\hello-world\hello-world.hdev succeeded.
Watching for directory changes or key presses in the console window... Press 'ESC' to exit.
Hello World!
caught an unhandled exception while executing "P:\projects\hdevrunner\hdevrunner\examples\hello-world\hello-world.hdev": proc name: "main (P:\projects\hdevrunner\hdevrunner\examples\hello-world\hello-world.hdev)"
Wrong value of control parameter 1" prog line: 9 type: "HDevEngineException"

write_string (FileHandle, 'Hello World!\n')
get_system ('halconversion', HalconVersion) <-----Wrong value of control parameter 1
write_string (FileHandle, 'You are using HALCON ' + HalconVersion + '\n')
close_file (FileHandle)

Execution of program P:\projects\hdevrunner\hdevrunner\examples\hello-world\hello-world.hdev failed. Watching for directory changes or key presses in the console window... Press 'ESC' to exit.
Watching for directory changes or key presses in the console window... Press 'ESC' to exit.
caught an unhandled exception while executing "P:\projects\hdevrunner\hdevrunner\examples\hello-world\hello-world.hdev": proc name: "main" proglineName: "" error code: 4
there is an invalid program line: 10" prog line: -1 type: "HDevEngineException"

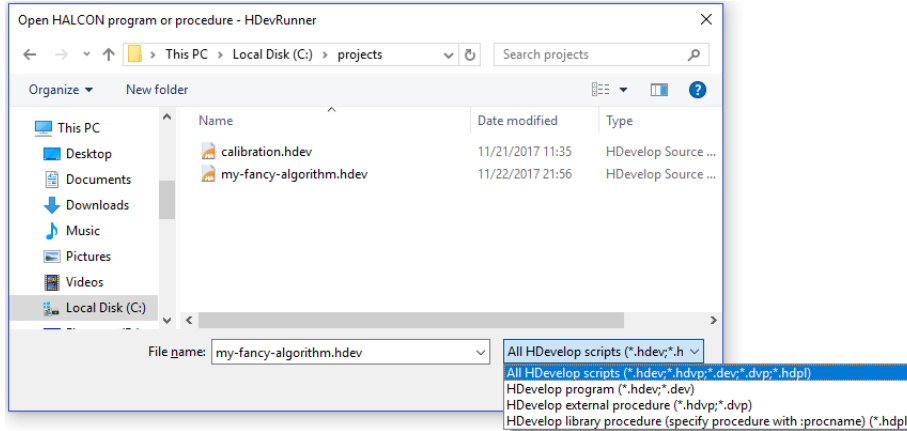
get_system ('version', Version)
write_string (FileHandle, 'You are using HALCON ' + HalconVersion + '\n') <-----
close_file (FileHandle)

Execution of program P:\projects\hdevrunner\hdevrunner\examples\hello-world\hello-world.hdev failed. Watching for directory changes or key presses in the console window... Press 'ESC' to exit.
Hello World!
You are using HALCON 12.0
Execution of program P:\projects\hdevrunner\hdevrunner\examples\hello-world\hello-world.hdev succeeded. Watching for directory changes or key presses in the console window... Press 'ESC' to exit.
```

# Features of HDevRunner Standard Edition

## Quickstart

As HDevRunner is intended to be automated on the customer machine, it is controlled mainly via command line. Nevertheless, as a convenience, HDevRunner can be started directly from the Windows Start Menu without any arguments and a file selection dialog appears. In this dialog, you can select a HDevelop script, HDevelop External Procedure, or a HDevelop Library Procedure to be executed. After the execution is finished, HDevRunner shows a dialog



In this dialog you can select a file exactly as you would specify it as a command line argument:

## Run HDevelop scripts

To run a HDevelop script, select a .hdev or .dev file in the file selection dialog (e.g. C:\Projects\MyScript.hdev) and click the *Open* button.

## Run HDevelop External procedures

Because of the flexibility of the approach, using HDevelop External procedures is recommended for using with HDevrunner. This way, during development in HDevelop, you could always put some initializing commands in a wrapper script that you load in HDevelop and then run the external procedure itself exactly as HDevRunner will do. For example, the script `examples/eval-on-multiple-images/scripts/eval-on-multiple-images.hdev` shows how HDevelop runs HDevRunner for the procedure `FindTriangles ()` in Watch Mode before it enters the procedure `FindTriangles ()` itself (e.g. to be edited). To run a HDevelop External procedure, select a .hdvp or .dvp file in the file selection dialog (e.g. C:\Projects\procedures\MyProcedure.hdvp) and click the *Open* button.

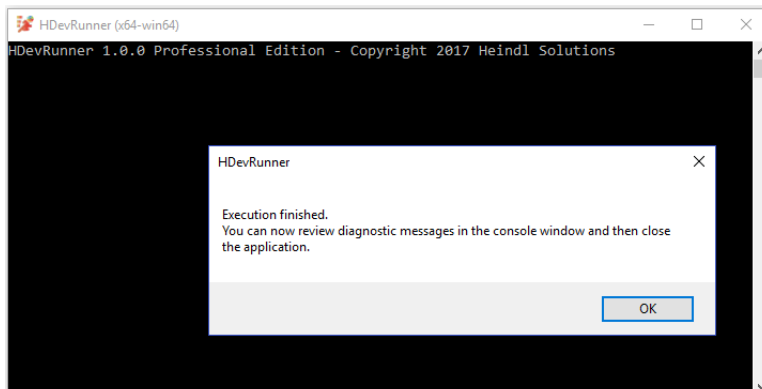
Please note that the procedure must not have any input parameters when used with HDevRunner Standard Edition. In the HDevRunner Professional Edition, you can specify control input parameters on the command line, but then its number must be equal to the number of control input parameters of the procedure. See again `examples/eval-on-multiple-images/procedures/FindTriangles.hdvp` for an example.

## Run HDevelop Library procedures

Library procedures are very similar to External procedures. You have to use the *colon syntax* to specify the library name followed by the library procedure name within the library. Example: `C:\Projects\MyLibrary.hdp1:SomeProc`

## Execution Finished

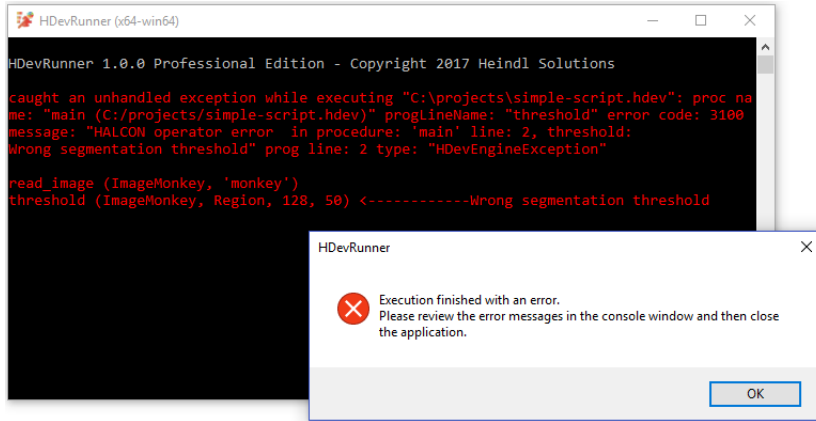
After execution is finished, the following dialog will be shown:



This gives you a chance to review any diagnostic or error messages or the time required for execution in the console before the application closes. If this is not desired, you could either run HDevRunner with the command line argument `--no-dialogs` or put the `exit ()` operator as last operator in your HDevelop script:

```
...
disp_continue_button (WindowHandle)
wait_seconds (2)
clear_calib_data (CalibDataID)
exit ()
```

If there is an error during execution, an error dialog will be shown and the errors can be reviewed in the console:



Note how the error line is annotated with the long arrow `<-----` followed by the message of the error that occurred in this line (here: *Wrong segmentation threshold*).

## Run with command line arguments

HDevRunner is most often run from the command line (or e.g. a Windows Shortcut or a Batch File) with command line arguments. To get you going quickly, start the *HDevRunner Command Prompt* from the Windows Start Menu. On startup it will show a short help for all supported command line arguments. You can start the 64-bit version of HDevRunner from there by typing

```
hdevrunner
```

If you have installed the 32-bit version of HDevRunner (and the corresponding x86sse2-win32 architecture of HALCON) you should use

```
hdevrunner32
```

instead. The following examples always assume that `hdevrunner` is used. Change it to `hdevrunner32` if you have installed the 32-bit version of HDevRunner.

To get a list of supported command line arguments execute:

```
hdevrunner --help
```

The possible command line arguments are described in the following sections.

### Minor and simple arguments

In this section, some simple arguments like `--help` or `--verbose` are described in the following table:

Argument (long form)	Argument (short form)	Description
<code>--help</code>	<code>-h</code>	Show help about the available command line arguments
<code>--verbose</code>	<code>-v</code>	Be more verbose in output in the console window
<code>--version</code>	<code>-V</code>	Output HDevRunner version information and exit
<code>--search-parent</code>		When relative paths are used, search parent directories for scripts or procedures. For example, <code>hdevrunner --search-parent myproc.hdev</code> searches the current working directory and all parent directories until a file named <code>myproc.hdev</code> is found
<code>--no-dialogs</code>		Never show any dialog (e.g. don't show the File open dialog or the Execution finished dialog)
<code>--no-default-window</code>		Do not open a default Graphics Window on startup
<code>--procedures-path &lt;path&gt;</code>		Add path to the procedures path, i.e. to the search path for HDevelop external procedures and HDevelop libraries. Use <code>--procedures-path</code> again for each additional path.
<code>--ignore-unresolved-lines</code>		Ignore unresolved script lines (e.g. external procedures that could not be found). If this option is not present, an error there is an unresolved procedure call might occur if a procedure could not be found and HDevRunner will exit with an error.

Argument (long form)	Argument (short form)	Description
--ignore-invalid-lines		Ignore invalid script lines (e.g. syntax errors in the script), i.e. code execution skips invalid lines. If this option is not present, an error there is an invalid program line might occur and HDevRunner will exit with an error.
--execute-procedures-jit-compiled		Try to run HDevelop procedures Just-In-Time compiled
--firstrun		Open "First Run" dialog (again) and exit
--get-hostids		Get HALCON host ids. Lists only the dongle host ids that are useful for licensing of HDevRunner. Note that you can also use the <i>License Center for HDevRunner</i> from the Windows Start menu for licensing with an easy-to-use graphical interface.
--get-version		Get currently active HALCON version
--activate-license <key>		Activate this product with a given key Note that you can also use the <i>License Center for HDevRunner</i> from the Windows Start menu for licensing with an easy-to-use graphical interface.

## Choose which HALCON version to use

If you have installed one or multiple HALCON version on a computer, there is always a default HALCON installation. This is specified in the environment variables HALCONROOT, HALCONARCH, HALCONEXAMPLES, and HALCONIMAGES. The HALCON bin/ directory is also added to the PATH environment variable by the HALCON installer, e.g.

```
PATH=C:\HALCON\HALCON-12.0\FLEX1m\x64-win64;C:\HALCON\HALCON-12.0\bin\x64-win64;C:\windows; ...
```

HDevRunner uses this information to run HDevelop scripts with the default HALCON installation. With the command line argument --halconversion you can ask HDevRunner to start with the HALCON settings as stored in the Windows Registry for the provided version. This allows for a quick switch to a secondary HALCON installation, e.g. to try if the developed scripts run with other versions of HALCON as well. For example

```
hdevrunner --halconversion 13.0 "C:\Projects\SomeScript.hdev"
```

tells HDevRunner to use the installed HALCON 13.0 instead of the default HALCON installation which might be HALCON 12.0 to run SomeScript.hdev. You can also specify only the start of the version number, e.g. only the major version number 13.

You can also specify which HALCON version to use by manually defining HALCONROOT with the command line argument --halconroot. Example:

```
hdevrunner --halconroot "C:\HALCON\HALCON-12.0" "C:\Projects\SomeScript.hdev"
```

## HDevEngine Remote Debugging

Starting with HALCON 13.0, HDevEngine Remote Debugging is available with HALCON. You can connect HDevelop to a process running HDevEngine (and thus connect to HDevRunner) on the same computer or remotely via network to start debugging the scripts that are running in this process. See the HALCON documentation for further details. To enable HDevEngine Remote Debugging you have to start HDevRunner with the argument --debug and optionally one of the following arguments

Argument	Description
--debug-password <password>	Allow remote debugging connections only if password <i>password</i> is given in HDevelop Remote Debugging connection dialog. If this argument is not present, no password will be used.
--debug-port <port>	HDevEngine Remote Debugging port (default is 57786)
--debug-wait-for-connection <true false>	HDevEngine Remote Debugging: Specify true to wait for a connection before execution starts (default is false)



# Features of HDevRunner Professional Edition

## Quickstart

The HDevRunner Professional Edition provides additional features on top of the HDevRunner Standard Edition features as described in the [previous chapter](#). Some of these features are especially useful for HALCON developers, like the Watch and the Check Mode. Other features like the Fullscreen Mode are handy if HDevRunner runs on your customer machines.

### Watch Mode

Watch mode is a feature to reload your scripts and procedures fast whenever a file changes, is added, or is removed on disk. It is typically used while editing a HDevelop script or procedure or a resource like test images.

HDevRunner Professional Edition is started in Watch Mode with the command line argument `--watch` (short `-w`). It will add the directory of the executed script or procedure and recursive its subdirectories to the list of watched directories automatically. Also, the HALCON `==import==` statements are parsed and the imported directories are added to the watched directories automatically. If you want to watch additional directories specify them with the command line argument `--watch-dirs <path>`. If you want to use multiple additional directories, specify `--watch-dirs` more than once, e.g.:

```
hdevrunner --watch --watch-dirs "C:/projects/images/good" --watch-dirs "C:/projects/images/bad" C:/projects/my-algorithm.hdevp
```

Now if you add new test images to `C:/projects/images/` and you have some test code like

```
list_image_files ('C:/projects', 'png', 'recursive', ImageFiles)
for Index := 0 to |ImageFiles|-1 by 1
    read_image (Image, ImageFiles[Index])
    * ...
endfor
```

your added images will automatically be included in the tests.

### Typical workflow

A typical example workflow in using Watch Mode might look like this:

- Start HDevelop
- Create an external procedure `my-algorithm.hdevp`
- Write some starter code like

```
dev_display (Image)
edges_sub_pix (Image, Edges, 'canny', 1, 20, 50)
dev_display (Edges)
```

- Open HDevRunner Command Prompt and run `my-algorithm.hdevp` with HDevRunner by executing:

```
hdevrunner --watch my-algorithm.hdevp
```

- Go back to HDevelop and tweak your algorithm, e.g. set the parameter `Filter` of `edges_sub_pix` to `sobel`:

```
dev_display (Image)
edges_sub_pix (Image, Edges, 'sobel', 1, 20, 50)
dev_display (Edges)
```

This way, you get immediate feedback whether your tweaks improved the overall performance of the vision algorithm or not.

### Check Mode

If Check Mode is enabled, HDevRunner does only a syntactic check on the script or external procedure provided on the command line and on its dependencies. Check Mode is enabled on the command line with `--checkonly` (short `-c`).

Check Mode allows for a fast check whether your code could run with other HALCON versions when combined with `--halconversion`.

You could for example write a batch file like

```
hdevrunner --halconversion 11.0 --checkonly some-procedure.hdevp hdevrunner --halconversion 12.0 --checkonly some-procedure.hdevp hdevrunner --halconversion 13.0 --checkonly some-procedure.hdevp
```

Each time the batch file is executed a check on valid syntax for HALCON 11, 12, and 13 is performed. As an example, a new operator available with HALCON 13.0 only could be used in the source code which would trigger error messages in the checks with HALCON 11 and 12.

You can even combine `--checkonly` with `--watch` to run the checks after each save to disk automatically, see [Watch Mode](#).

### Fullscreen Mode

If HDevRunner Professional Edition is run in Fullscreen Mode, all Graphics Windows will be stretched to the full screen size. No other GUI elements of the Windows operating system (e.g. taskbar) will be visible. This is an ideal mode for presentations or when running your code on customer machines as there will be no distraction from your contents.

Fullscreen Mode is enabled with the command line argument `--fullscreen` (short `-f`).

Normally, HDevRunner starts with a default Graphics Window at Row/Column (0/0) with a size of (640/480). The visible part will be set to Upper/Left (0/0) and Lower/Right (479/639). This is the same behavior as in HDevelop. If using Fullscreen Mode, the part will be kept the same, but the window size will be the full size of the screen.

In Fullscreen Mode, you will not see the HDevRunner Debugging Console window. You can switch to the console by pressing Alt+Tab (maybe multiple times). From there, you can quit HDevRunner by clicking on the cross as you would do with any other Windows application.