

APPLICATION NOTE

Trigger over Ethernet - Action Commands

V2.0.1

Scope of this document

This document covers the following topics:

- Benefits of Action Commands (also known as Trigger over Ethernet or ToE)
- Options for sending out Action Commands
- Using Action Commands with Vimba X Viewer and Vmb APIs
- Advanced scenarios, Troubleshooting

Benefits of Action Commands

Single cable solution

Action Commands are also known as Trigger over Ethernet (ToE), since no additional trigger cable is required. You can send the trigger signal to one or multiple cameras.

High-precision trigger broadcast

In contrast to a software trigger, which is the established Trigger over Ethernet method, Action Commands enable broadcasting a trigger signal to multiple GigE cameras at almost the same time. Therefore, Action Commands are more precise than a software trigger.

Flexible multi-camera applications

To enhance the flexibility of your vision application, Allied Vision GigE cameras support two different Action Commands: Action0 and Action1 are provided. Additionally, you can define camera groups and assign cameras to one or multiple sub-groups.

Fast reaction

Trigger signals sent out from the host PC always have some latency caused by the PC's operating system. If your vision application requires a particularly fast reaction to a trigger signal, you can send out Action Commands with special hardware.

Compatible with GigE Vision

Action Commands are covered by the GigE Vision Specification and therefore are supported by a large selection of machine vision hardware and software.

Prerequisites

To use Allied Vision cameras and Action Commands, you need:

- Allied Vision GigE camera(s) with Action Commands support. To find out if your camera supports Action Commands, see the technical manual or the camera model webpage on <https://www.alliedvision.com>.
- Vimba, Vimba X, or special hardware capable of sending out Action Commands, or a GenICam-compliant third-party software.

This document assumes you have already installed and configured the host adapter and applicable drivers according to the instructions in the User Guide for your Allied Vision GigE camera.

Recommendation: Alvium cameras with the latest firmware and the Vimba X SDK offer best support for Action Commands.

Additional information

We recommend downloading the following documents:

- The **Features Reference** for your camera provides a general overview of GigE camera features.
- The **User Guide or Technical Manual** for your camera contains camera-specific information and technical data.
- Vimba X: Read the Developer Guide, chapter Viewer Guide (included in the SDK).
- Vimba: Read the Vimba Viewer Guide (included in the SDK).



Download camera documentation

You can download camera documentation from:

<https://www.alliedvision.com/en/support/technical-documentation.html>

Options for sending out Action Commands

There are several options for sending out Action Commands.

- You can send out Action Commands from the host PC with:
 - Vimba X Viewer (or Vimba Viewer)
 - Vimba X APIs (Vmb APIs, recommended for Alvium) or Vimba APIs
 - A third-party software with Action Commands support
- You can send out Action Commands from special hardware such as IMAGO Technologies VisionBox.

All options require these basic steps:

1. Open the camera(s).
2. Configure the trigger:
 - a. Set the trigger selector (e.g., FrameStart).
 - b. Set the trigger source (Action0 or Action1).
 - c. Set the trigger mode (On).

4. Configure the ActionControl parameters on each camera.
5. Configure the ActionControl parameters on the host PC or the special third-party hardware.
6. Start acquisition.
7. Send out an Action Command.

Configuring ActionControl parameters

The following ActionControl parameters must be configured on the camera(s) and then on the host PC.

- `ActionDeviceKey` must be equal on the camera and on the host PC. Before a camera accepts an Action Command, it verifies if the received key is identical with its configured key. Note that `ActionDeviceKey` must be set each time the camera is opened.
Range (camera and host PC): 0 to 4294967295
- `ActionGroupKey` means that each camera can be assigned to exactly one group for Action0 and a different group for Action1. All grouped cameras perform an action at the same time. If this key is identical on the sender and the receiving camera, the camera performs the assigned action.
Range (camera and host PC): 0 to 4294967295
- `ActionGroupMask` serves as a filter that specifies which cameras within a group react to an Action Command. It can be used to create sub-groups. To react to an Action Command, the value in the camera must have one bit in common with the received value from the sender. You can define different Group Masks for Action0 and Action1.
Range (camera): 0 to 4294967295
Range (host PC): 1 to 4294967295

Using Action Commands with Vimba X or Vimba Viewer

Vimba X Viewer: See the Developer Guide, section Viewer Guide in Vimba X directory on your host PC or the [online version](#).

Vimba Viewer: see the Viewer Guide in the Vimba directory on your host PC or the [online version](#).

Programming Action Commands



Loading camera settings with an API

If you have configured the camera settings for Action0 with Vimba X or Vimba Viewer, you can conveniently load them with a Vmb or Vimba API. To do this, save the setting in the Viewer and then see the description in the API manuals.

Vimba X users: The Vimba X SDK currently contains an Action Commands example for the Python API. You can find an example in C on GitHub: https://github.com/alliedvision/Vimba_X_examples.

Please add the example to the directory of the Vimba X examples provided with the SDK.

Vimba users: The Vimba SDK contains Action Commands examples for all APIs.

Advanced scenarios

Scenario A: Using Action0 and Action1

You can use `Action0` and `Action1` to trigger different kinds of actions such as starting and stopping image acquisition. If you send out Action Commands with Vimba X Viewer, the camera automatically toggles between `AcquisitionStart` and `AcquisitionEnd` each time an Action Command is received.

Example:

For `Action0` and `Action1`, set a common `ActionDeviceKey` (for example, `ActionDeviceKey = 1`)

For `Action0`, use `SelectorAcquisitionStart` and, for example:

```
ActionGroupKey = 1
ActionGroupMask = 1
```

For `Action1`, use `SelectorAcquisitionEnd` and, for example:

```
ActionGroupKey = 1
ActionGroupMask = 1
```

Scenario B: Assigning cameras to different groups

You can use `ActionGroupKey` to assign each camera to one group within `Action0` and optionally a different group within `Action1`. Assigning one camera to multiple groups within the same Action is not possible.

Example:

`SelectorFrameStart` and `Action0` are used for all cameras. As soon as an Action Command is sent, all assigned cameras acquire one frame.

Camera1 and Camera2 are members of ActionGroup 1:

```
ActionDeviceKey = 1
ActionGroupKey = 1
ActionGroupMask = 1
```

Camera3 and Camera4 are members of ActionGroup 2:

```
ActionDeviceKey = 1
ActionGroupKey = 2
ActionGroupMask = 1
```

Scenario C: Assigning cameras to sub-groups

You can use `ActionGroupMask` as a filter for sub-groups. A camera can be assigned to one or multiple sub-groups or it can be excluded from all of them. If the Group Mask on the host PC and on the camera have one or more bits with value 1 in common, then the camera reacts to an Action Command.

Examples of binary Group Mask values (for simplification, the example shows four digits):

Sent Group Mask value	Camera Group Mask value	Camera reaction
1111	0001	Yes
1101	0010	No
1111	0011	Yes
1110	0011	Yes

Table 1: Examples of binary Group Mask values

Cameras can be assigned to one, multiple, or no sub-groups:

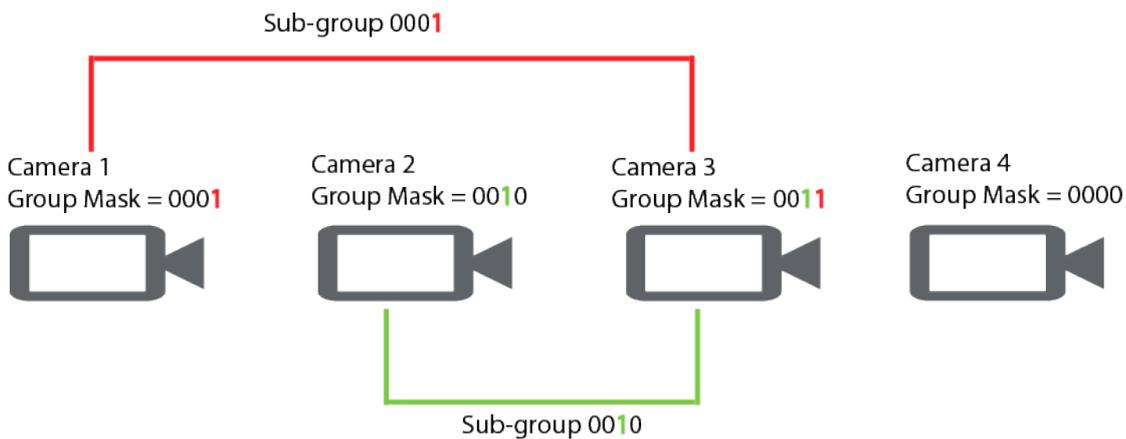


Figure 6: Group Mask principle: A common bit 1 = a common sub-group

While Group Mask value 0 is valid on a camera (this particular camera is currently excluded from all groups and sub-groups and ignores Action Commands), sending out an Action Command with value 0 is not allowed per the GigE Vision Specification: It causes an error because no camera will react to it. Therefore, the Group Mask range on the PC starts with 1, while the range on the camera starts with 0.

Combining scenarios

You can combine the scenarios as required. For example, you can use `Action0` and `Action1`, assign the cameras to different groups, and use Group Mask to create additional sub-groups.

Tips and troubleshooting

- If Action Commands are unavailable although your camera should support them, download the latest firmware from: <https://www.alliedvision.com/en/support/firmware.html>
- `ActionDeviceKey` must be set each time the camera is opened and is read-only. Do not close the camera while working with Action Commands. Note that there is no warning if the maximum value is exceeded, as it is read-only.
- Vimba X and Vimba Viewer: Click the Freerun button before sending out Action Commands.
- If you use an Ethernet router, make sure all cameras are in the same subnet. Using a switch normally does not affect Action Commands.
- Be aware that the camera verifies if the received key matches its configured key in the following order:
 1. `ActionDeviceKey`
 2. `ActionGroupKey`
 3. `ActionGroupMask`
- Per the GigE Vision Specification, Action signals can only be asserted if the device has an open primary control channel or when unconditional action mode is enabled. (Note that unconditional action mode is not supported by all devices.)
- For troubleshooting, use `EventAction0` and `EventAction1` (see GigE Features Reference or Alvium Features Reference if you have an Alvium camera).