

Basler pylon

```
// Create an instant camera object with the first
Camera_t camera( CTIFactory::GetInstance().Creat

// Register an image event handler that accesses
camera.RegisterImageEventHandler(=new CSampleImage
Ownership_TakeOwnership);

// Open the camera.
camera.Open();
```

APPLICATION NOTE

How to use the Basler pylon GenTL Producers for Basler GigE and USB 3.0 Cameras with MathWorks MATLAB

Document Number: AW001343

Version: 03 Language: 000 (English)

Release Date: 6 November 2017

Contacting Basler Support Worldwide

Europe, Middle East, Africa

Basler AG
An der Strusbek 60–62
22926 Ahrensburg
Germany

Tel. +49 4102 463 515
Fax +49 4102 463 599

support.europe@baslerweb.com

The Americas

Basler, Inc.
855 Springdale Drive, Suite 203
Exton, PA 19341
USA

Tel. +1 610 280 0171
Fax +1 610 280 7608

support.usa@baslerweb.com

Asia-Pacific

Basler Asia Pte. Ltd.
35 Marsiling Industrial Estate Road 3
#05–06
Singapore 739257

Tel. +65 6367 1355
Fax +65 6367 1255

support.asia@baslerweb.com

www.baslerweb.com

All material in this publication is subject to change without notice and is copyright Basler AG.

Table of Contents

1	Introduction.....	2
2	Requirements.....	2
3	Installation.....	3
3.1	Installing the Basler pylon GigE and USB GenTL Producers	3
3.2	Installing and Configuring MATLAB	5
4	Operating a Basler USB3 Vision Camera with MATLAB.....	9
5	Operating a Basler GigE Camera with MATLAB.....	11
6	Troubleshooting	13
6.1	Troubleshooting USB 3.0 Cameras.....	13
6.1.1	Troubleshooting Using MATLAB	13
6.1.2	Troubleshooting Using the Basler pylon Camera Software Suite	15
6.2	Troubleshooting GigE Vision Cameras	16
6.2.1	Troubleshooting Using MATLAB	16
6.2.2	Troubleshooting Using the Basler pylon Camera Software Suite	18

1 Introduction

The Basler pylon GigE and USB GenTL producers enable you to operate Basler GigE Vision and USB3 Vision cameras with MATLAB (www.mathworks.com) or any software libraries that support the GenICam GenTL standard (<http://www.emva.org>) and implement their own GenTL consumer.

This document describes how to configure MATLAB so that Basler GigE and USB 3.0 cameras are detected in MATLAB. Once the cameras are displayed in MATLAB, you can configure them as desired and acquire images.

The Basler pylon GigE and U3V GenTL producers are included in the pylon Camera Software Suite 5.0.11 or higher which is available for Windows, Linux x86, and Linux ARM. For more information about the pylon Camera Software Suite and to download the latest version go to www.baslerweb.com.

2 Requirements

The procedures described in this document assume that the following software and hardware is present:

- MATLAB R2017a (9.2.0.556344) 64-bit with the Image Acquisition Toolbox version 5.2 (R2017a)
- Image Acquisition Toolbox Support Package for GenICam Interface version 17.2.0.0, which enables you to acquire video and images from GenTL-compliant cameras
- Image Acquisition Toolbox Support Package for GigE Vision Hardware version 17.2.0.0, which is required for advanced IP address configuration and troubleshooting of GigE cameras
- Basler pylon Camera Software Suite 5.0.11 including the pylon GigE and USB GenTL 64-bit producers
- Basler GigE and/or Basler USB 3.0 cameras
- GigE network card and/or USB 3.0 host controller card recommended by Basler
- GigE and or USB 3.0 cables recommended by Basler



The Basler pylon GenTL producers are compliant with GenTL version 1.5.

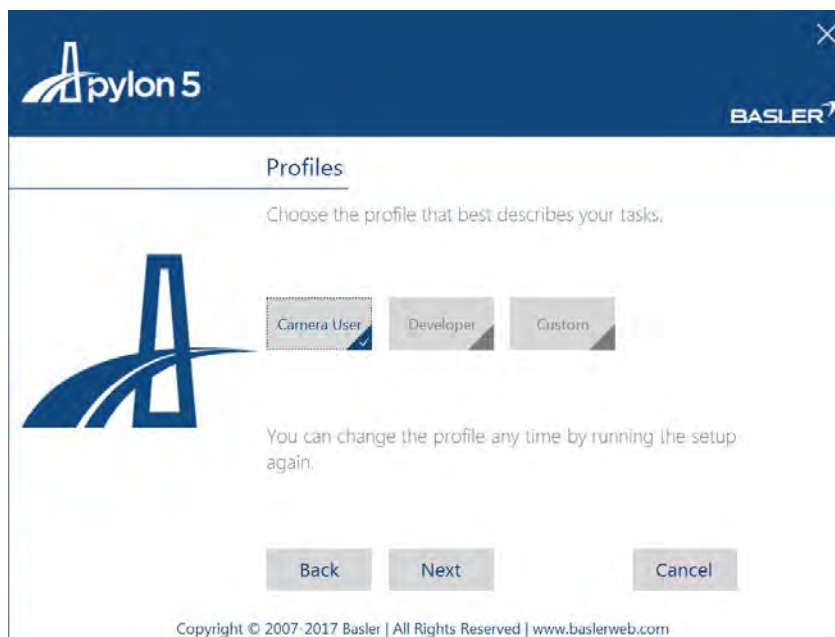
For more information about recommended accessories for Basler GigE and Basler USB 3.0 cameras, visit to the Basler website (www.baslerweb.com) or contact your local Basler Customer Service team.

3 Installation

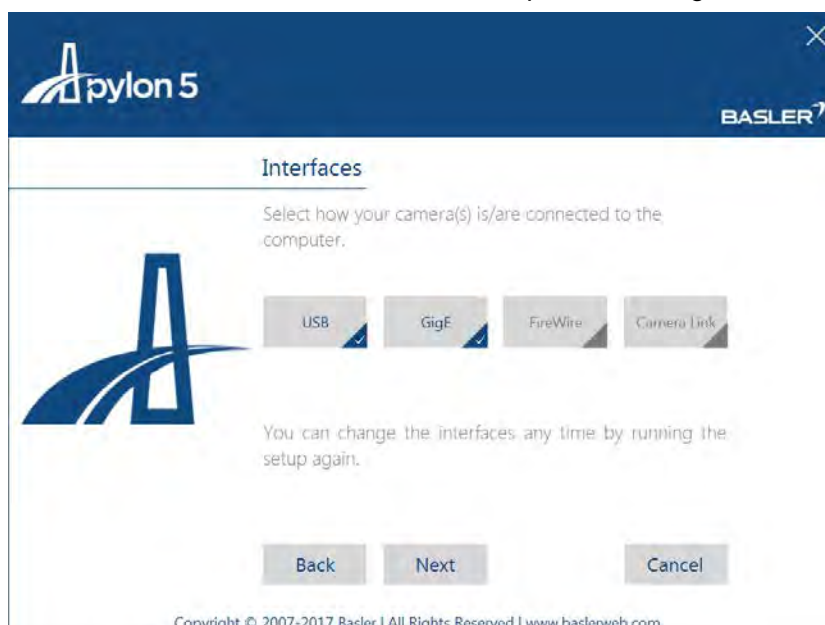
3.1 Installing the Basler pylon GigE and USB GenTL Producers

To install the Basler pylon GigE and USB GenTL producers:

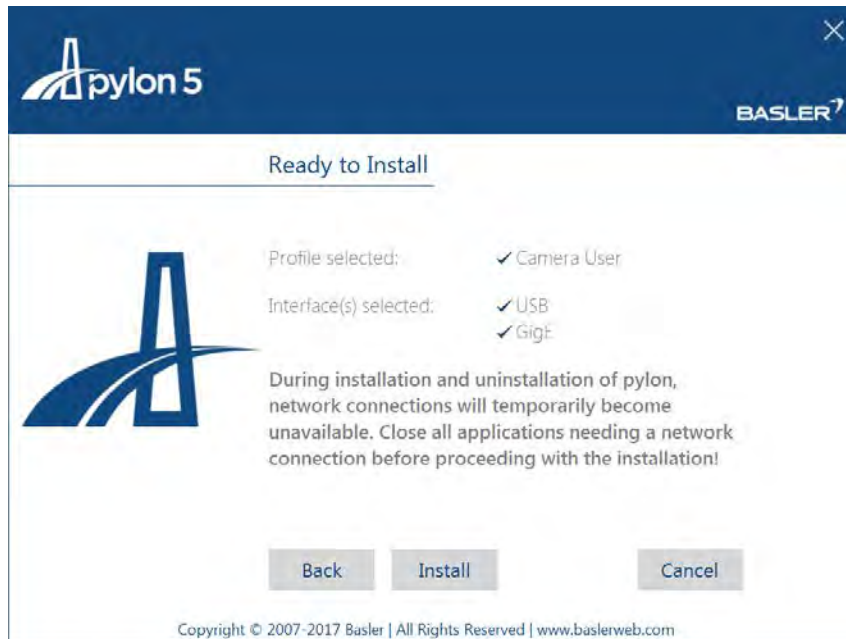
1. Run Basler pylon 5.0.11.xxx.exe.
2. On the **Profiles** page, choose either the **Camera User** or the **Developer** profile.
If you intend to develop applications based on the pylon Camera Software Suite APIs, choose the **Developer** profile.



3. Click **Next**.
4. On the **Interfaces** page, select the **Interfaces** for which you want to install the necessary drivers, runtime environment, and GenTL producers, e.g., USB and GigE.



5. Click **Next**.
6. Click **Install** to install the selected components now.

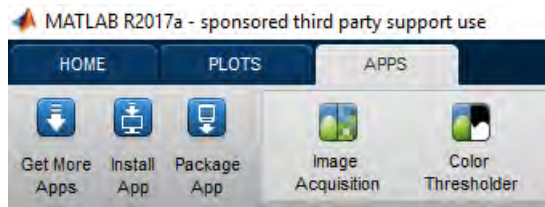


7. After the installation has completed, log off from your computer and then log in again. This is necessary for the changed system environment variables to take effect.

3.2 Installing and Configuring MATLAB

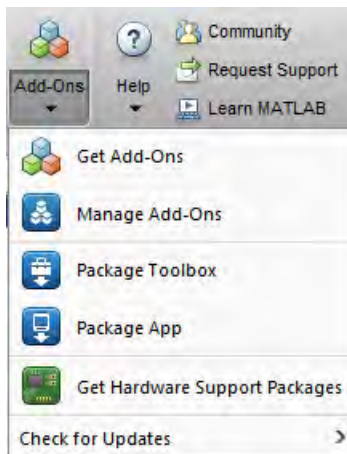
To install and configure MATLAB:

1. In MATLAB, go to the **APPS** tab and make sure that the **Image Acquisition** app is installed.

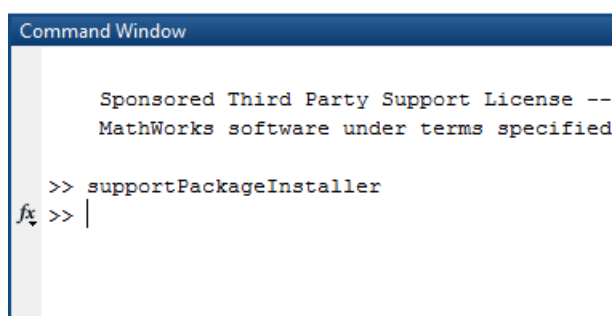


If the Image Acquisition app has not been installed yet, install it by running the Support Package Installer in one of the following ways:

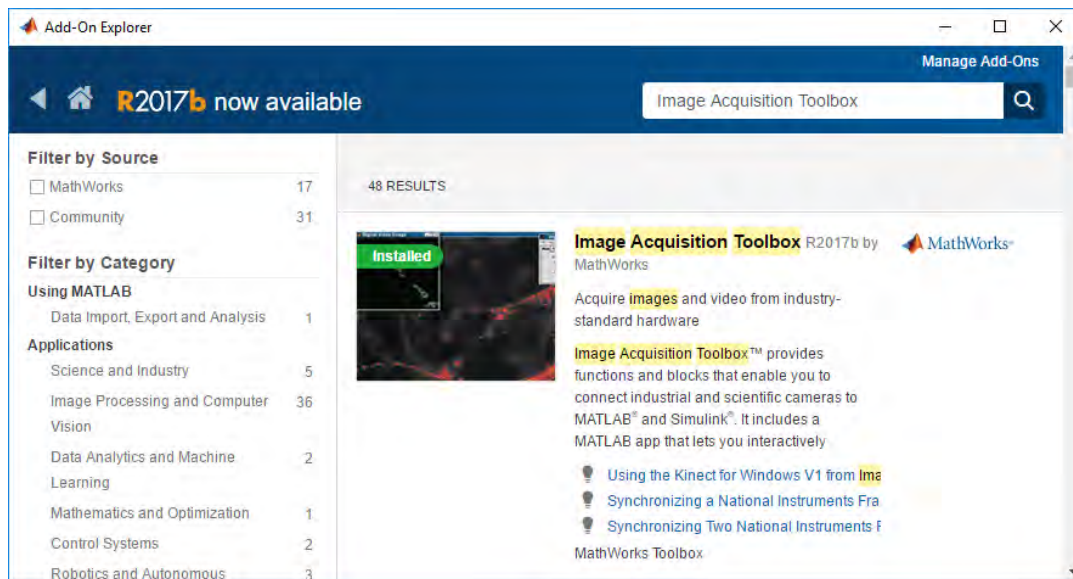
- On the **HOME** tab, click **Add-Ons > Get Hardware Support Packages**.



- In the MATLAB **Command Window**, enter:
`supportPackageInstaller`



- a. In the Add-On Explorer, remove the **Clear Filter** field and enter '**Image Acquisition Toolbox**'.



- b. Select the Image Acquisition Toolbox and install it.
2. Check if the MATLAB GenTL consumer is available. In the MATLAB **Command Window**, enter:

```
imaghwinfo
```

```
Command Window

Sponsored Third Party Support License -- for use
MathWorks software under terms specified in your

>> imaghwinfo

ans =

struct with fields:

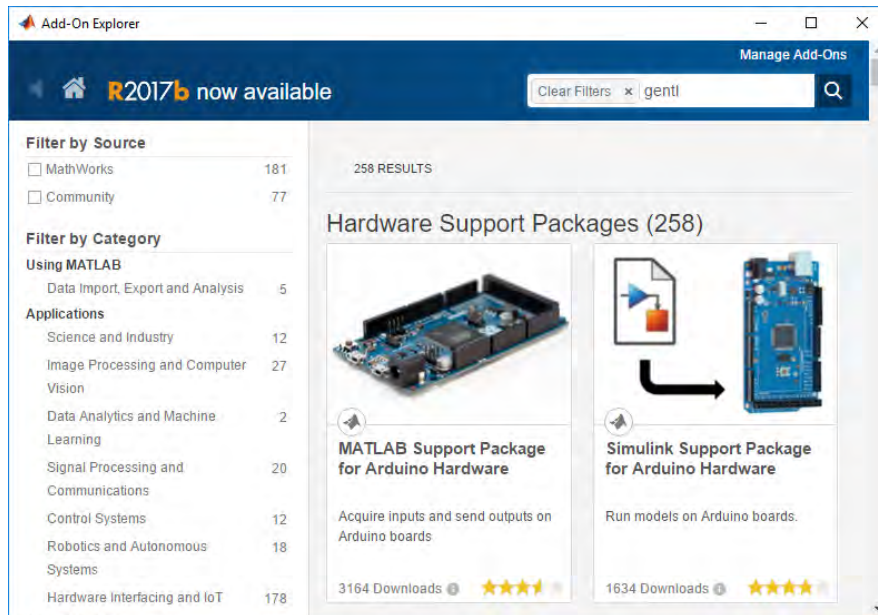
    InstalledAdaptors: {'gentl' 'gige'}
    MATLABVersion: '9.2 (R2017a)'
    ToolboxName: 'Image Acquisition Toolbox'
    ToolboxVersion: '5.2 (R2017a)'

fx >> |
```

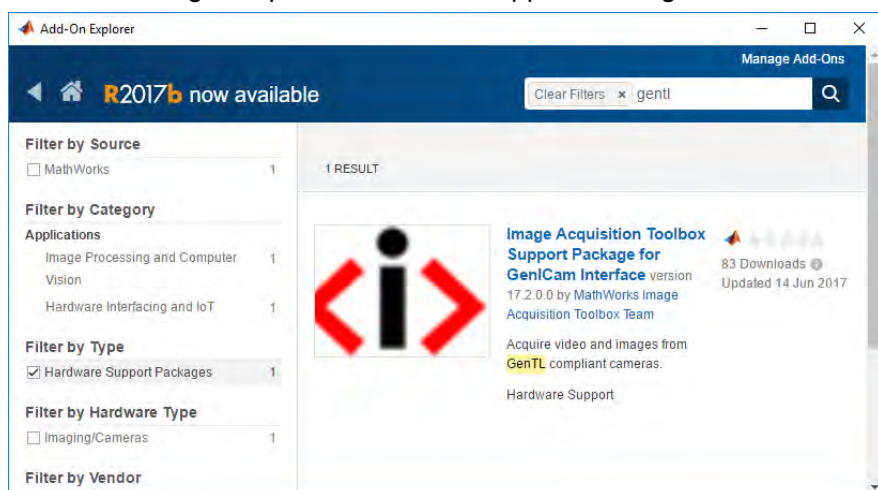
If '**gentl**' is not listed under **InstalledAdaptors**, install the MATLAB GenTL Consumer in one of the following ways:

- On the **HOME** tab, click **Add-Ons > Get Hardware Support Packages**.
- In the MATLAB **Command Window**, enter:
supportPackageInstaller

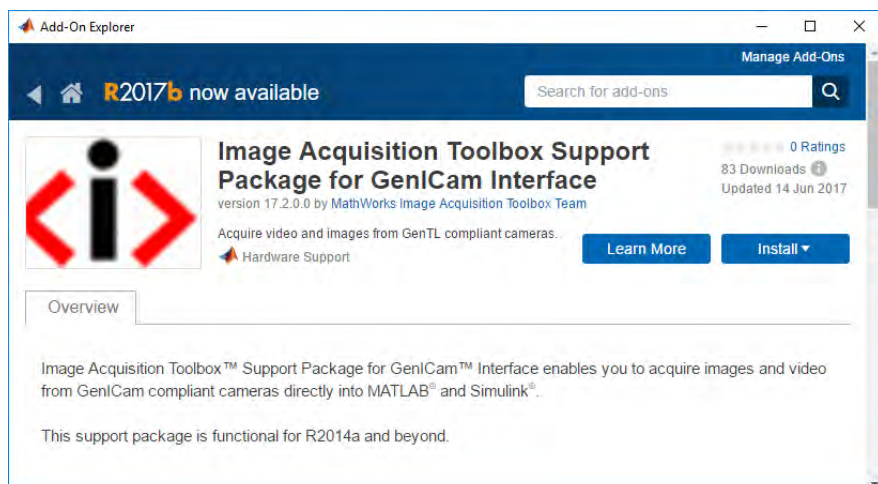
3. In the Add-On Explorer enter 'gentl' in the search field.



4. Select the Image Acquisition Toolbox Support Package for GenICam Interface.



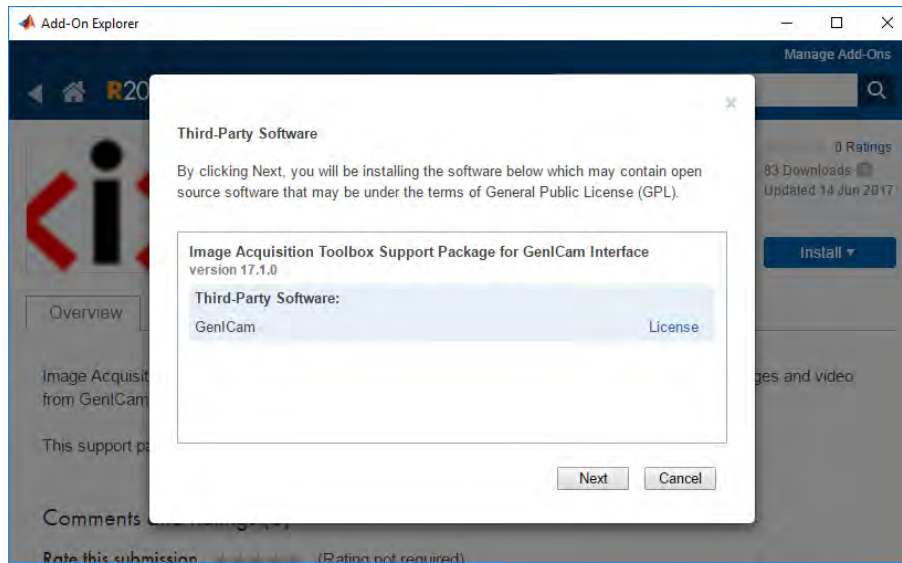
5. Click Install to install the Support Package for GenICam Interface.



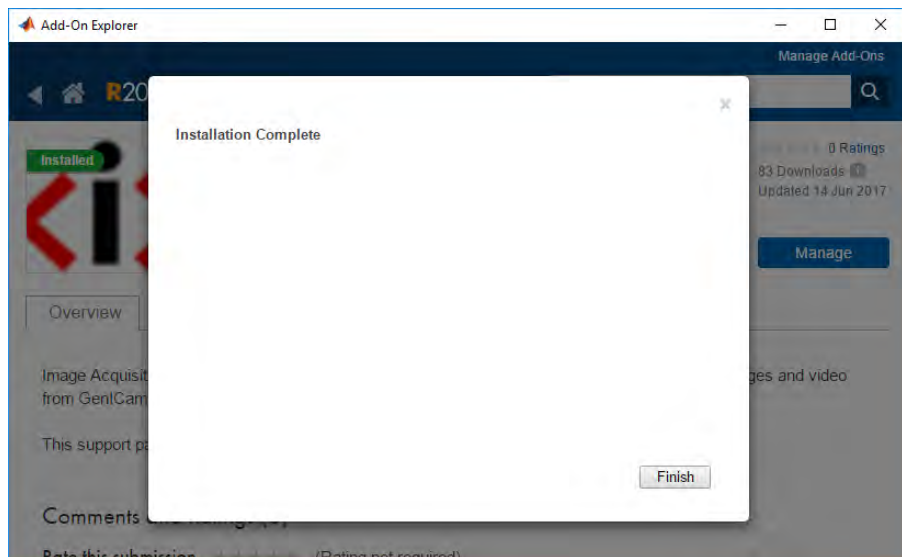
A window with licensing information will be displayed.

6. Accept the MathWorks Auxiliary Software License Agreement and the Genicam License Agreement.

7. Click **Next** to install the Support Package for GenICam Interface.



8. When the installation is complete, click **Finish**.



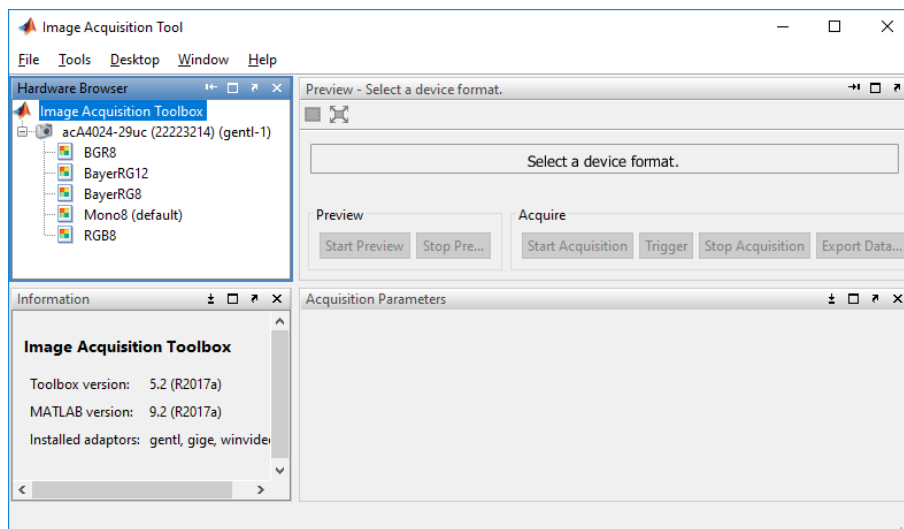
4 Operating a Basler USB3 Vision Camera with MATLAB

This section explains how to access a Basler ace USB 3.0 camera in MATLAB and how to configure the camera's features.

To access a Basler ace USB 3.0 camera in MATLAB:

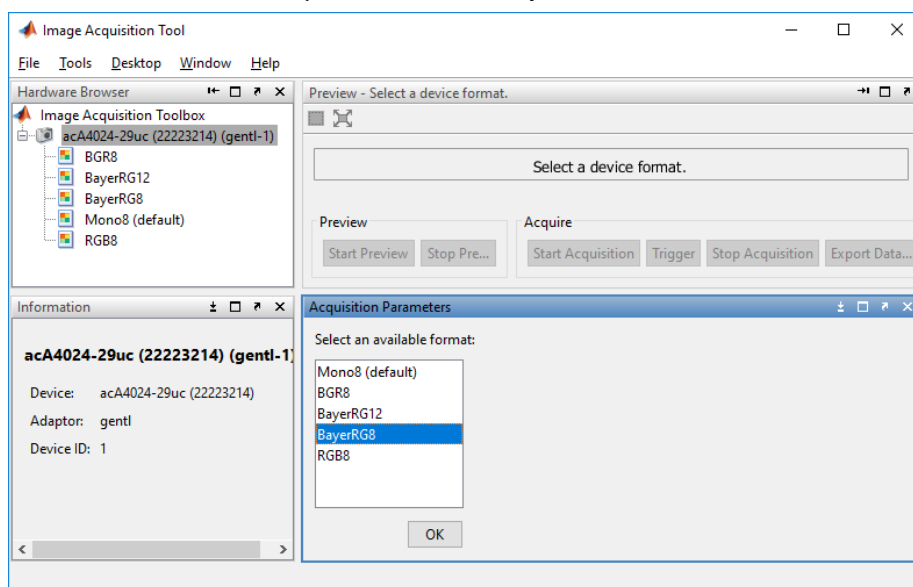
1. Connect your Basler ace USB 3.0 camera to a port of a USB 3.0 host controller card recommended by Basler.
2. In MATLAB, click **APPS > Image Acquisition**.

The **Image Acquisition Tool** opens in a new window. In this example, a Basler ace acA4024-29uc camera is used. It is listed in the **Hardware Browser** pane.

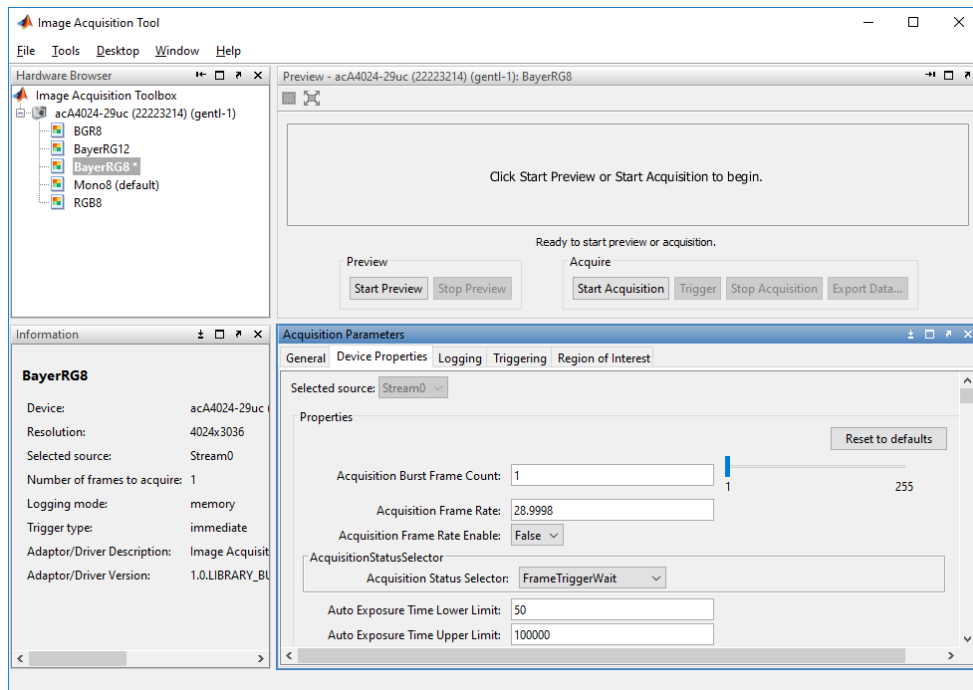


3. When you select the camera in the **Hardware Browser** pane, the available pixel formats will be listed in the **Acquisition Parameters** pane.
4. Select the desired pixel format and click **OK**.

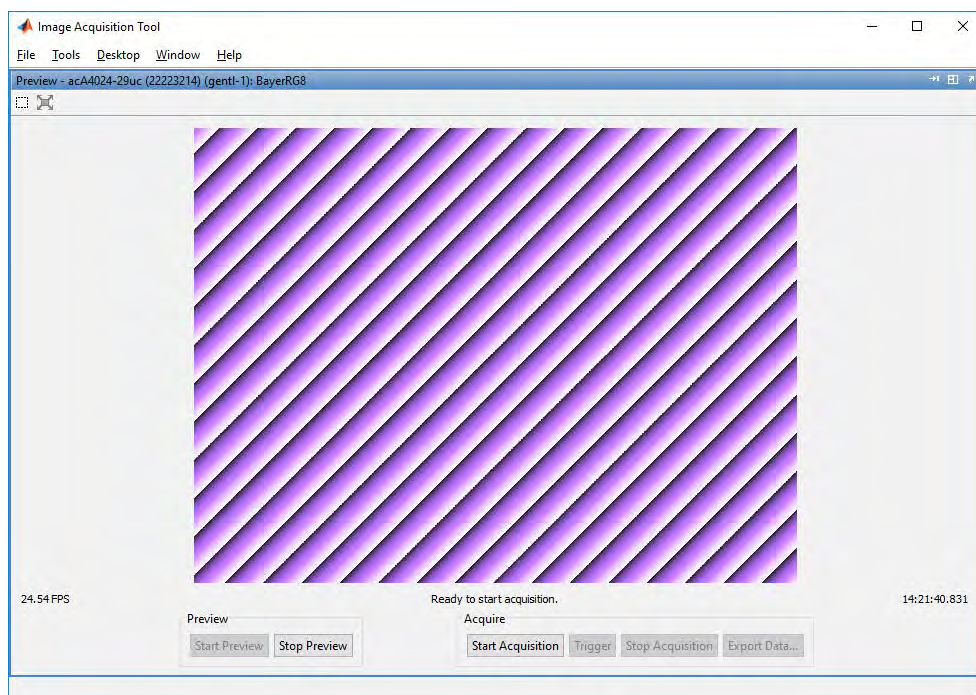
The camera has been opened and is ready for use now.



5. In **Acquisition Parameters** pane, go to the **Device Properties** tab to access all currently supported camera features.



6. In the **Preview** pane, click **Start Preview** in order to get a live image from the camera.



5 Operating a Basler GigE Camera with MATLAB

This section explains how to access a Basler ace GigE Vision camera in MATLAB and how to configure the camera's features.

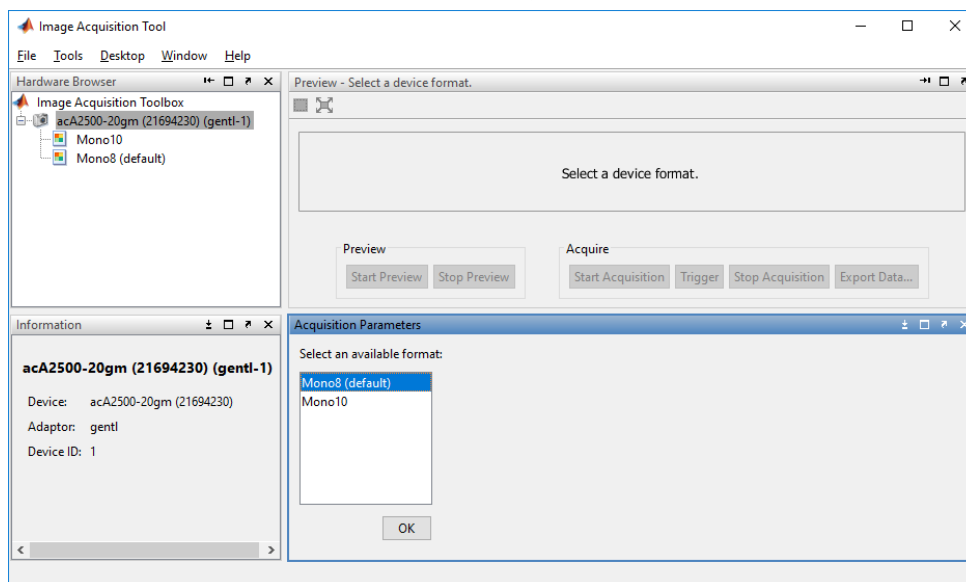
To access a Basler ace GigE camera in MATLAB:

1. Connect your Basler ace GigE camera to a GigE network card recommended by Basler.
2. In MATLAB, click **APPS > Image Acquisition**.

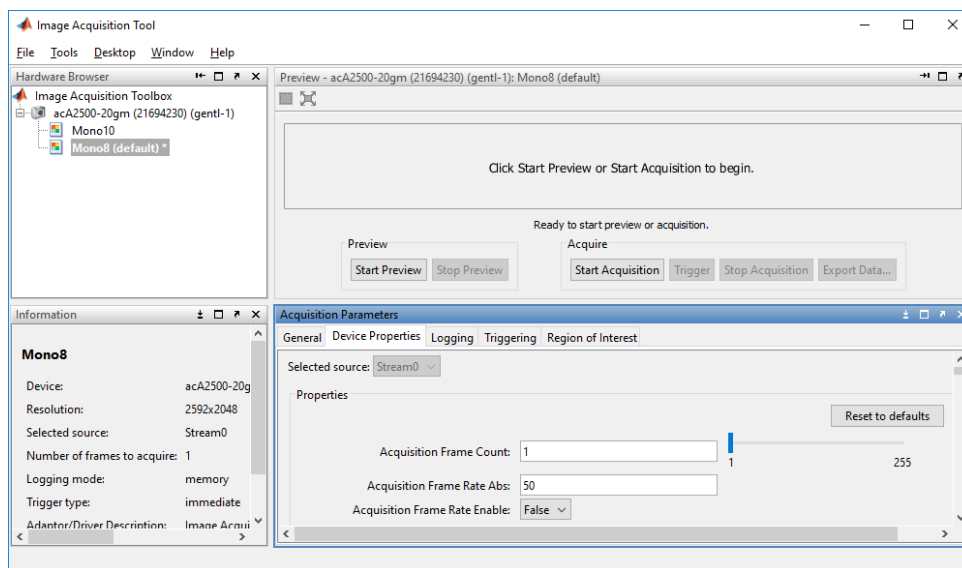
The **Image Acquisition Tool** opens in a new window. In this example, a Basler ace acA2500-20gm camera is used. It is listed in the **Hardware Browser** pane.

3. When you select the camera in the **Hardware Browser** pane, the available pixel formats will be listed in the **Acquisition Parameters** pane. Select the desired pixel format and click **OK**.

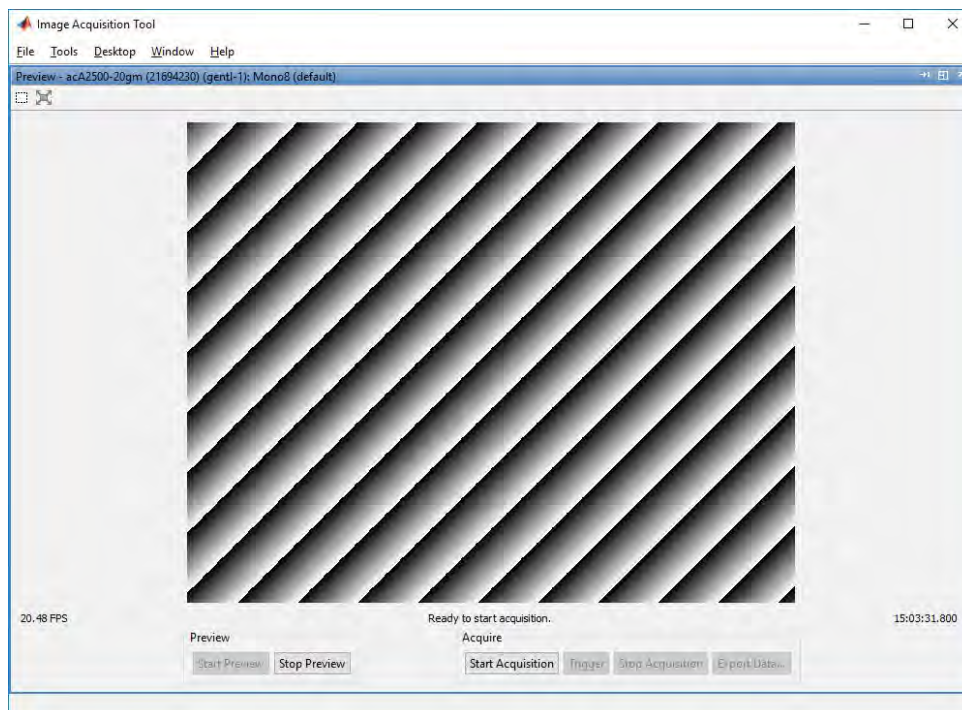
The camera has been opened and is ready for use now.



4. In the **Acquisition Parameters > Device Properties** tab you can access all currently supported camera features:



5. In the **Preview** pane, click **Start Preview** in order to get a live image from the camera:



6 Troubleshooting

6.1 Troubleshooting USB 3.0 Cameras

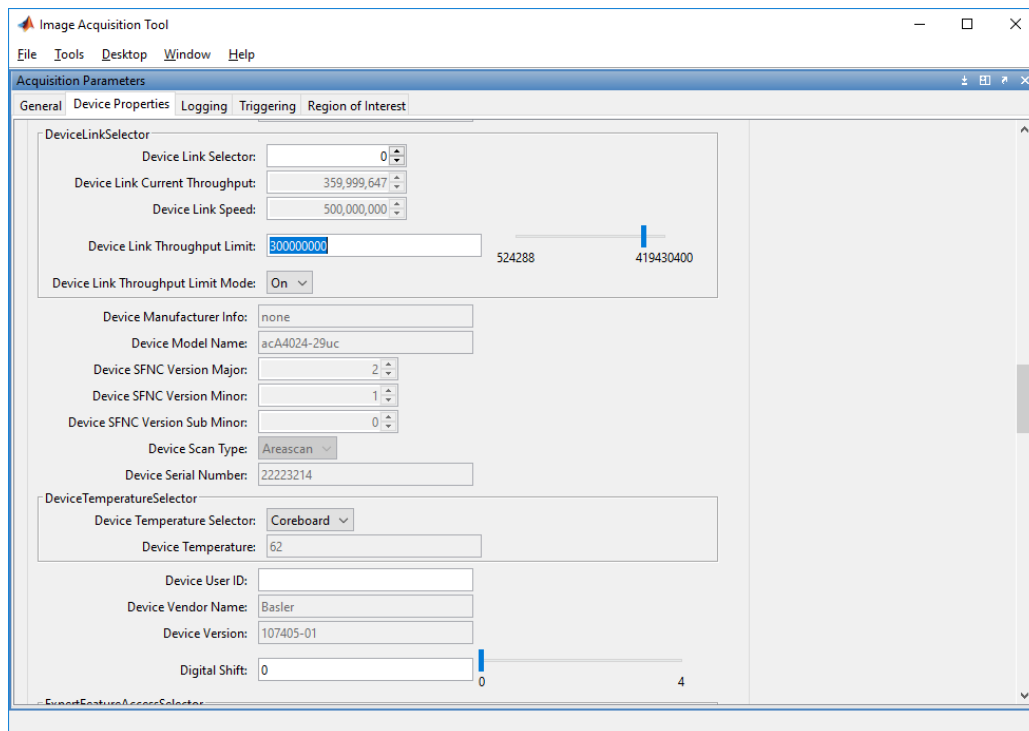
6.1.1 Troubleshooting Using MATLAB

If the images you acquire are corrupt, try decreasing the bandwidth currently used by the camera in the MATLAB Image Acquisition Tool. You can do this in two different ways:

- Image Acquisition Tool:

In the Image Acquisition Tool, go to **Acquisition Parameters > Device Properties** and make the following changes:

- Set Device Link Throughput Limit Mode to On.
- **Device Link Throughput Limit:** Change the value (in Bytes/s) until MATLAB stops acquiring corrupt images, e.g., 300 000 000 Bytes (i.e., 300 Mbytes).



- Command Window:

In MATLAB, click **HOME > Command Window** and enter the following:

```
vid= videoinput('gentl', 1, 'BayerRG8')
src= getselectedsource(vid);
src.DeviceLinkThroughputLimitMode = 'On'
src.DeviceLinkThroughputLimit = 300000000
```

```

Command Window

Sponsored Third Party Support License -- for use only to support products interfaced to
MathWorks software under terms specified in your company's restricted use license agreement.

>> vid= videoinput('gentl', 1, 'BayerRG8')

Summary of Video Input Object Using 'acA4024-29uc (22223214)'.

Acquisition Source(s): Stream0 is available.

Acquisition Parameters: 'Stream0' is the current selected source.
                        10 frames per trigger using the selected source.
                        'BayerRG8' video data to be logged upon START.
                        Grabbing first of every 1 frame(s).
                        Log data to 'memory' on trigger.

Trigger Parameters: 1 'immediate' trigger(s) on START.

Status: Waiting for START.
        0 frames acquired since starting.
        0 frames available for GETDATA.

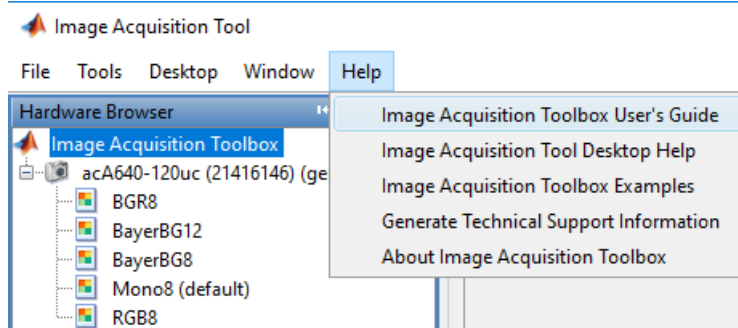
>> src= getselectedsource(vid);
>> src.DeviceLinkThroughputLimitMode = 'On'

```

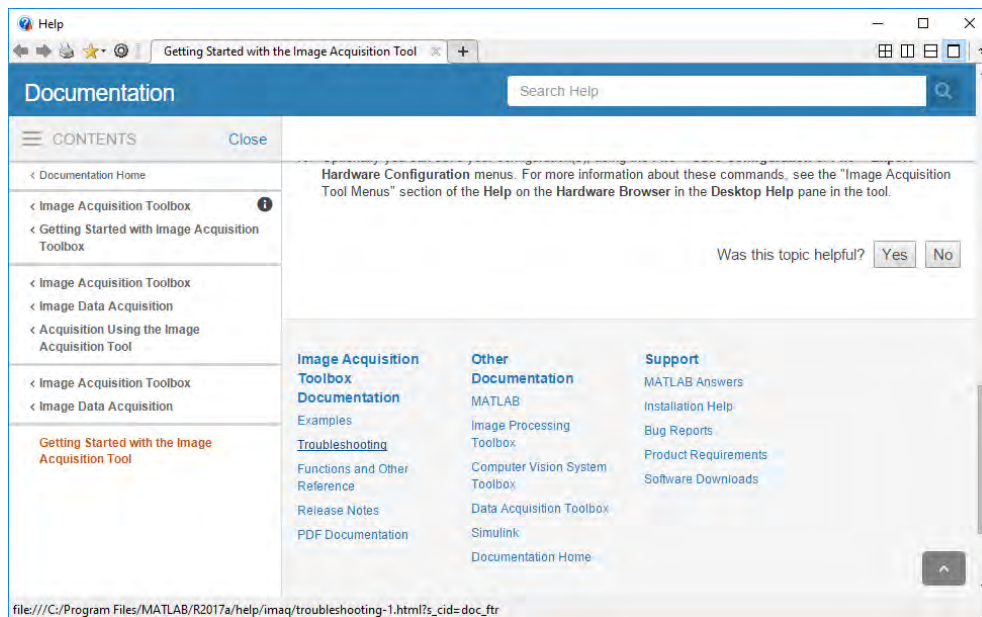
If decreasing the value of the **Device Link Throughput Limit** parameter doesn't improve the image acquisition issue, you may need further debugging with the Basler pylon Viewer, which is part of the Basler pylon Camera Software Suite.

For further MATLAB troubleshooting information, you have the following options:

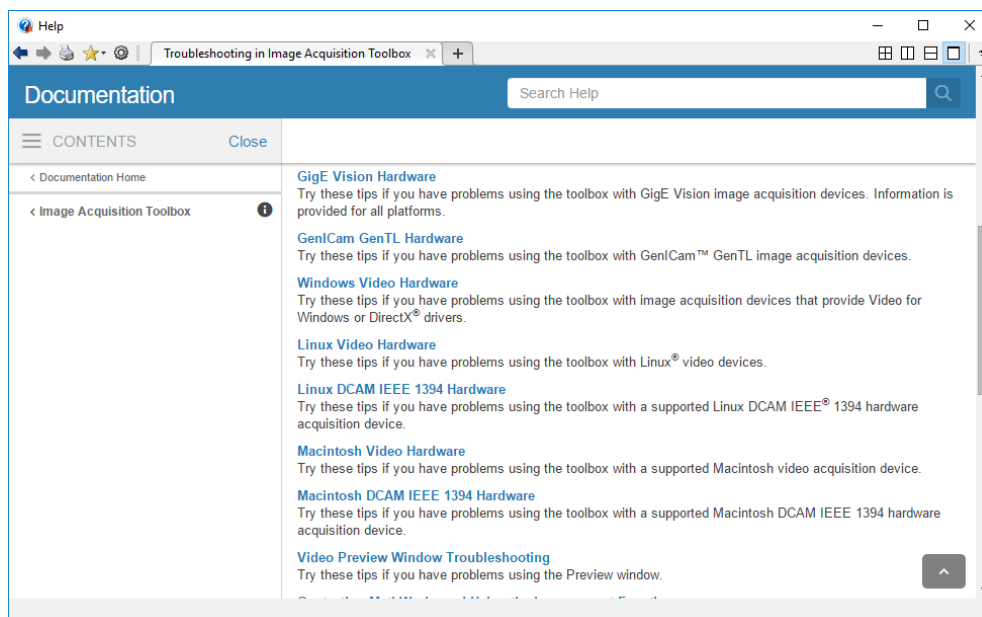
- Image Acquisition Toolbox User's Guide:
 1. In MATLAB, click **APPS > Image Acquisition > Help > Image Acquisition Toolbox User's Guide**.



2. Scroll down and select **Troubleshooting**.



3. Here, you will find troubleshooting tips related to the GenICam GenTL Hardware:



- Contact the MATLAB technical support team.

6.1.2 Troubleshooting Using the Basler pylon Camera Software Suite

The Basler pylon Camera Software Suite offers the following tools: pylon Viewer, pylon USB Bandwidth Manager and pylon USB Configurator.

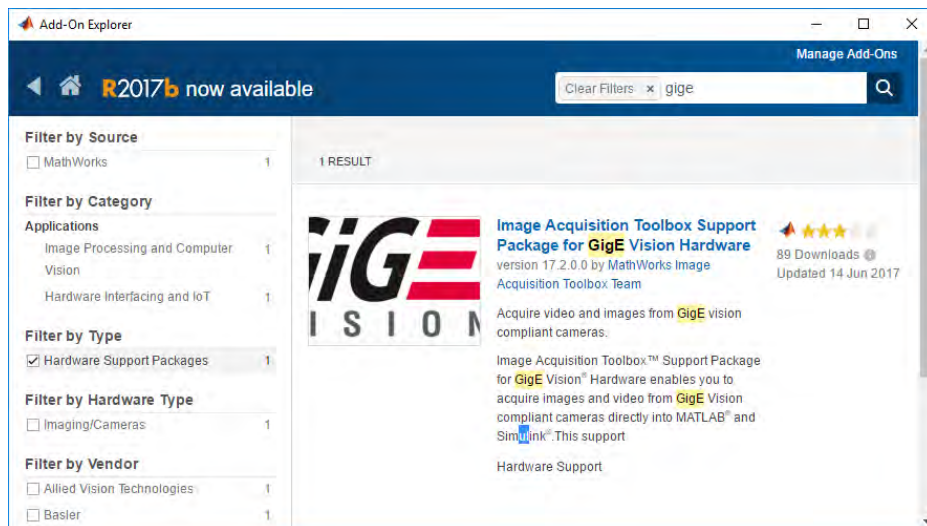
If you need help using these tools, contact your local Basler Customer Service Team (www.baslerweb.com).

6.2 Troubleshooting GigE Vision Cameras

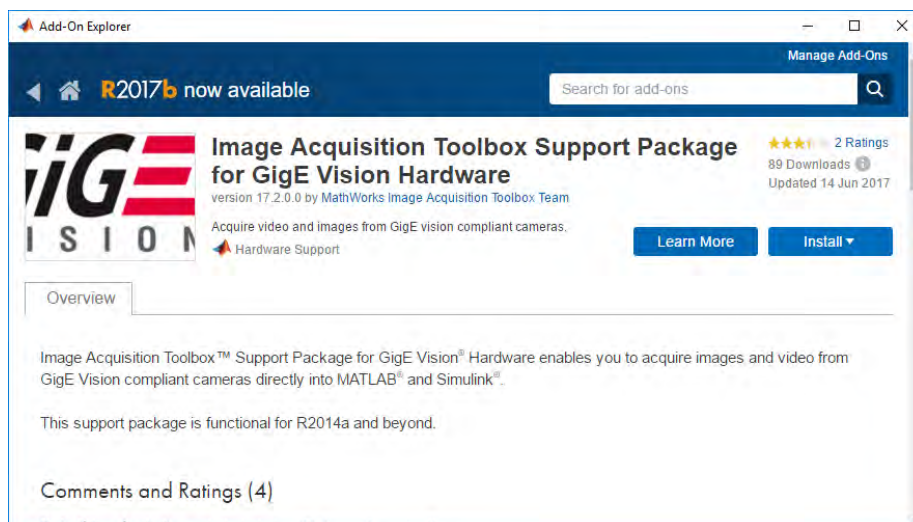
6.2.1 Troubleshooting Using MATLAB

If your GigE Vision camera is not shown in the MATLAB Image Acquisition Toolbox or the images you acquire are corrupt, you have to install the **Image Acquisition Toolbox Support Package for GigE Vision Hardware**. To do so, follow these steps:

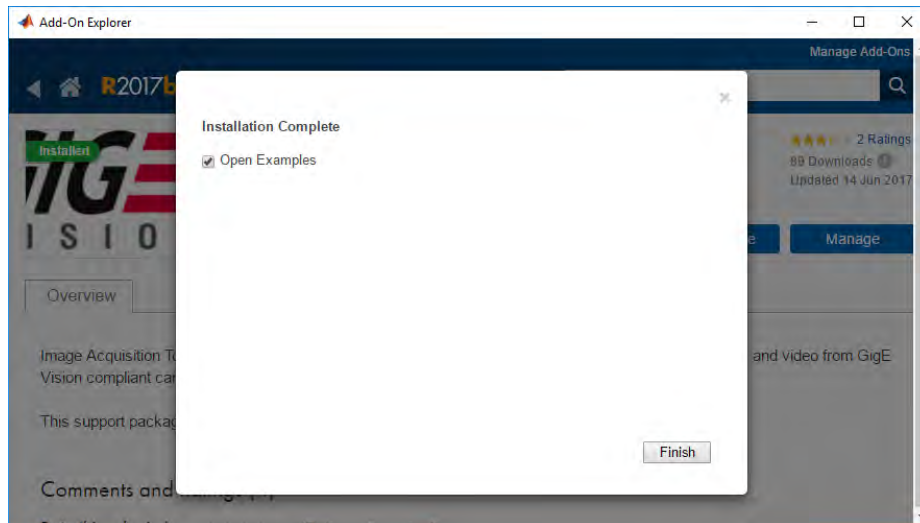
1. On the **HOME** tab, click **Add-Ons > Get Hardware Support Packages**.
2. In the Add-On Explorer enter 'gige' in the search field:



3. Select the **Image Acquisition Toolbox Support Package for GigE Vision Hardware** and click **Install** to install it.

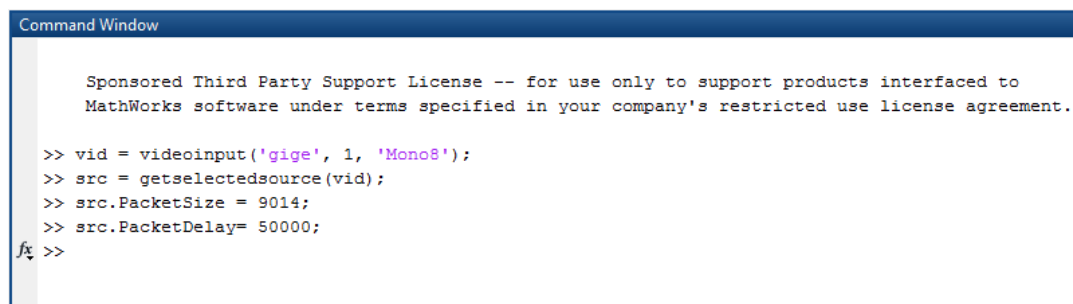


4. When the installation process has completed, click **Finish**.



5. In MATLAB, click **HOME** > **Command Window** enter the following to optimize the **Packet Size** and the **Packet Delay** parameters.

```
vid= videoinput('gige', 1, 'Mono8')
src= getselectedsource(vid);
src.PacketSize = 9014;
src.PacketDelay= 5000;
```



When using jumbo frames for the **Packet Size** camera parameter, i.e., values above 1500 Byte, make sure that your network adapter supports that value. Otherwise, image acquisition may fail.

Increasing the **Packet Delay** camera parameter, may reduce the camera's resulting maximum acquisition frame rate.

For more information about troubleshooting GigE Vision cameras in MATLAB, refer to the following documents:

- MATLAB GigE Vision Quick Start Configuration Guide:
https://www.mathworks.com/matlabcentral/answers/uploaded_files/41167/GigEVisionQuickStart.pdf
- MATLAB FAQ:
<https://de.mathworks.com/matlabcentral/answers/91834-how-do-i-calculate-the-packet-delay-for-a-gige-vision-camera-to-prevent-dropped-frames>

6.2.2 Troubleshooting Using the Basler pylon Camera Software Suite

The Basler pylon Camera Software Suite offers the following tools: pylon Viewer, pylon GigE Bandwidth Manager and pylon IP Configurator.

If you need help using these tools, contact your local Basler Customer Service Team (www.baslerweb.com).

Revision History

Document Number	Date	Changes
AW00134301000	4 Feb 2015	Initial release version of this document.
AW00134302000	6 Nov 2017	Updated the document to reflect the use of MATLAB R2017a. Updated the document to reflect the use of pylon 5.0.11. Added Chapter 5 to reflect the operation of GigE Vision cameras with MATLAB. Added Chapter 6.2 to reflect the troubleshooting of GigE Vision cameras.
AW00134303000	6 Nov 2017	Corrected a typo.