See the possibilities

Getting Started Guide

JAI SDK

Software Development Kit and Control Tool

Document Version: K
Document P/N: 10671

See the possibilities

Table of Contents

1 General

This document takes you through the process of downloading and installing the SDK & Control Tool software, as well as optimizing the Ethernet settings on your PC for optimal streaming of video data from the JAI GigE Vision® and GenTL™ cameras.

The JAI SDK & Control Tool is fully compliant with the GenTL™, GigE Vision® and GenICam™ standards, and can therefore be used with all GenTL™ compliant devices as well as all GigE Vision® compliant cameras from JAI.

For further information on the GenTL™, GigE Vision® and GenICam™ standards see www.machinevisiononline.org and www.genicam.org

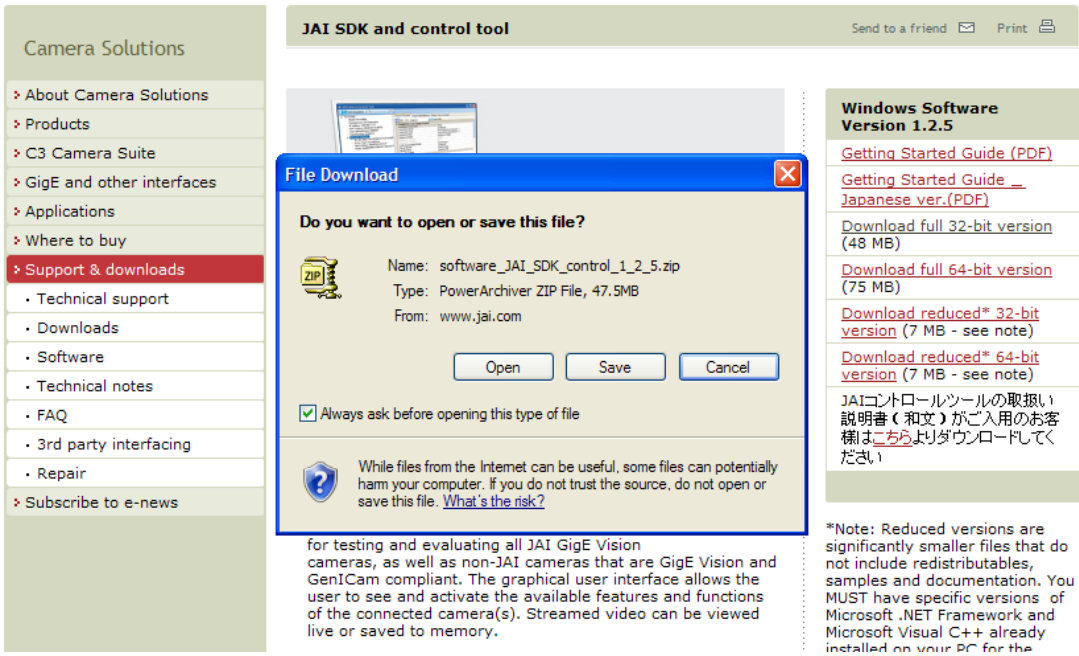
2 Downloading the software

The latest version of the JAI SDK & Control Tool software is available, free of charge, at: http://www.jai.com/en/support/jai_sdk_and_control_tool.

There are four different versions available:

- Full 32-bit (x86) version for Windows XP, Windows Vista, and Windows 7/8.
- Full 64-bit (x64/AMD64) version for Windows XP, Windows Vista, and Windows 7/8.
- Reduced 32-bit version without .NET framework version 2.0 redistributables, samples and documentation
- Reduced 64-bit version without .NET framework version 2.0 redistributables, samples and documentation

Figure 1. File download screen



Click the save button and specify location on your computer.

You need at least 80 megabytes of free hard disk space on your computer in order to download and install the software (for the full 32-bit version).

3 Installing on your PC

Once the software (zip file) has been downloaded, you can run the installation. Either extract the files from the zip file, or run the installation directly from the zip folder.

Please note that the JAI SDK & Control Tool is based on Microsoft Windows, and will therefore only work on this operating system.

Note: Windows 8 and 8.1 - The installation program MUST be run as administrator (or the installation may fail).
Note: If a previous version of the JAI SDK & Control Tool is installed on the computer, please uninstall it before installing a newer version. In addition, after an uninstallation:

1. Delete any files in the "C:\Program Files\JAI" folder.
2. Delete the previous version of the filter driver ("C:\Windows\System32\drivers\JaiFilterDriver.sys"

3.1 JAI SDK Software Installer

The installer looks for several programs, including Microsoft .NET Framework version 2.0, Windows Installer 3.1, Microsoft Visual C++ 2005 and GenICam™ version 2 Redistributables. If these are not found on the system, the installer has the software included in the SDK, and provides an option to install the applications. Allow the installations of these prerequisite programs or the JAI SDK installation closes.

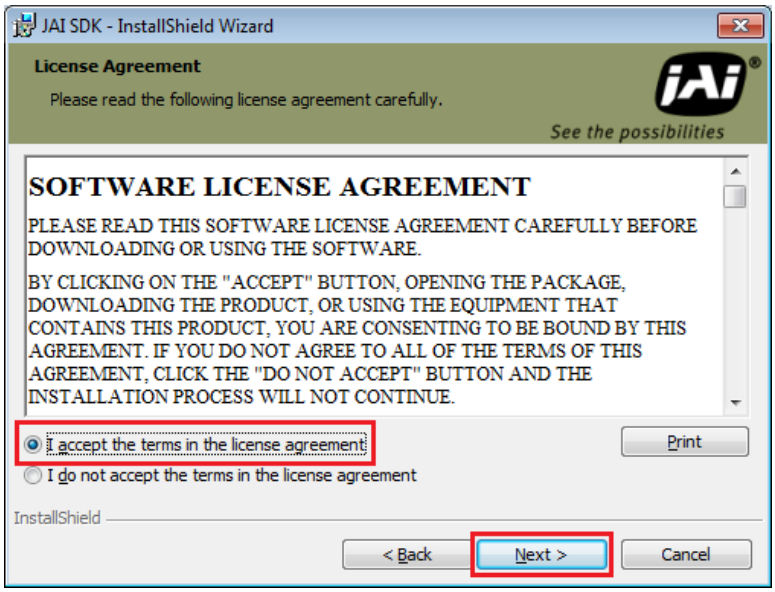
Figure 2. JAI SDK installation screen

After installing all the needed redistributables the welcome dialog for the JAI SDK installer will be displayed.

Figure 3. Installer welcome dialog

Read and accept the license agreement.

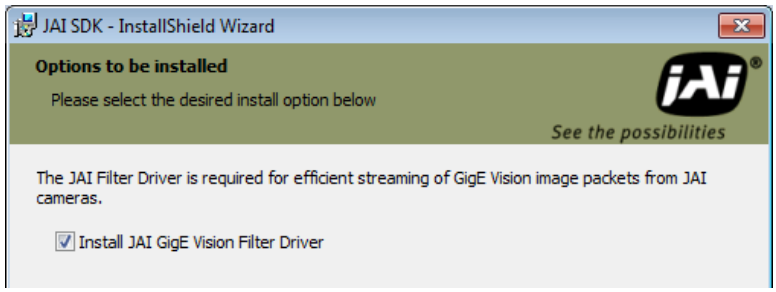
Figure 4. License Agreement



3.2 Install the JAI GigE Vision® Filter Driver

The software offers to install the JAI GigE Vision® Filter Driver. The performance of this driver is superior to the "Socket Driver", which always will be installed, and it is required for efficient streaming of the GigE Vision® image packets from the cameras. The Filter Driver can easily be uninstalled and reinstalled after the SDK installation so it is recommended to install it as default.

Figure 5. **Filter Driver Option**



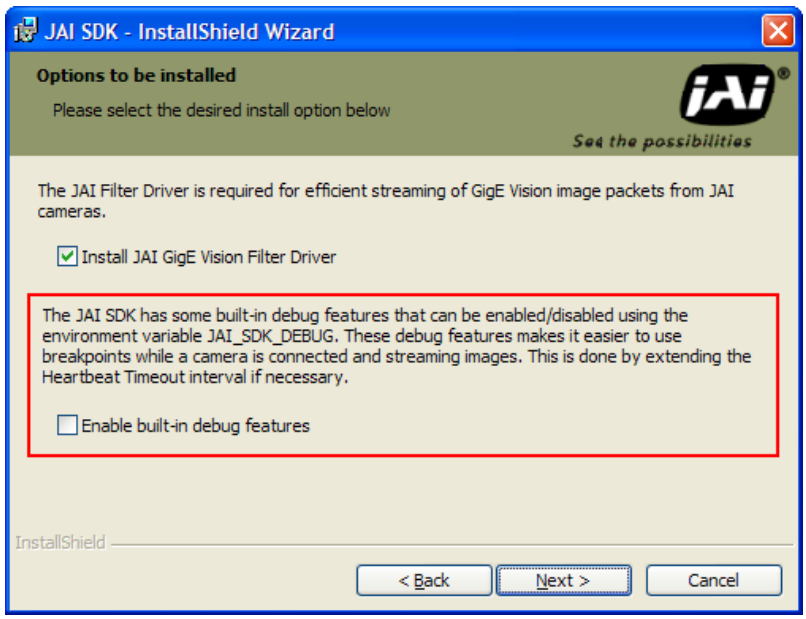
3.3 Enabling Transport Layer debug features

The installers make it possible to enable special Transport Layer debug features that are very helpful during application development using the JAI SDK. If the user hits break-points during a debug session then the connected GigE Vision® camera will likely timeout because the "Heartbeat" telegrams are not received within the configured Heartbeat Timeout period. The camera will then close the connection to the host PC (so other applications will be able to take control of the camera) and if the camera was streaming images then the stream channel will be closed as well inside the camera. This is a mandatory behavior for all GigE Vision® compliant cameras!

To avoid that the camera disconnects and potentially closes the image stream channel, the Transport Layer is capable of detecting "delays" that might have been caused by break-points. If this is detected then the Transport Layer will automatically extend the Heartbeat Timeout value inside the connected camera with a factor of 100 and thereby make it less likely that the camera times out. When the Transport Layer closes the camera connection then the original Heartbeat Timeout value will automatically be restored!

The debug feature is by default disabled but during installation the user can enable this feature by setting a checkmark in the "Enable built-in debug features" checkbox.

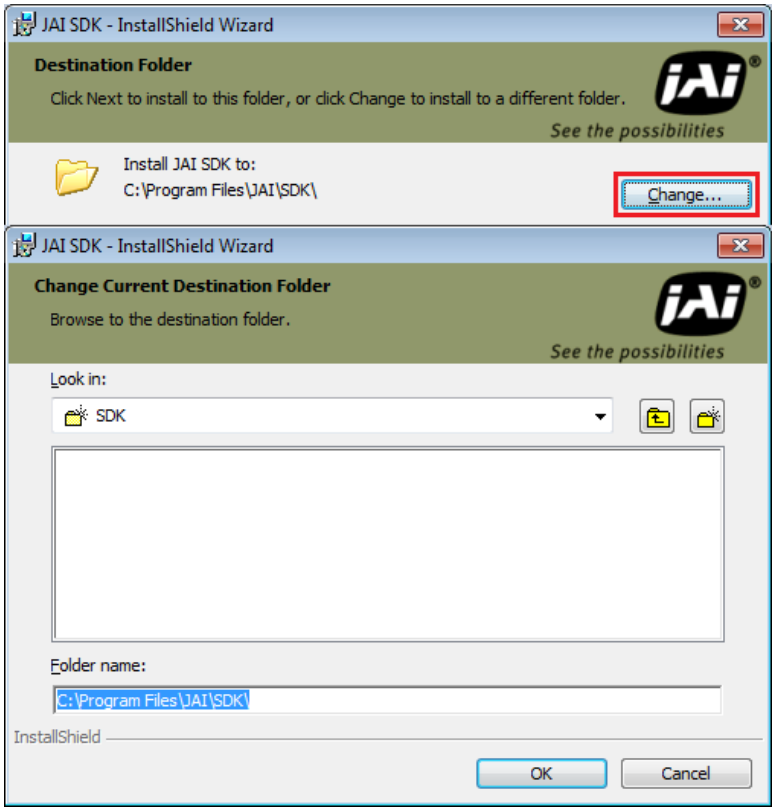
Figure 6. **Debug Option**



3.4 Software Installation Location

The software offers a default location <Program Files Directory>\JAI\SDK, but it is possible to select your own path.

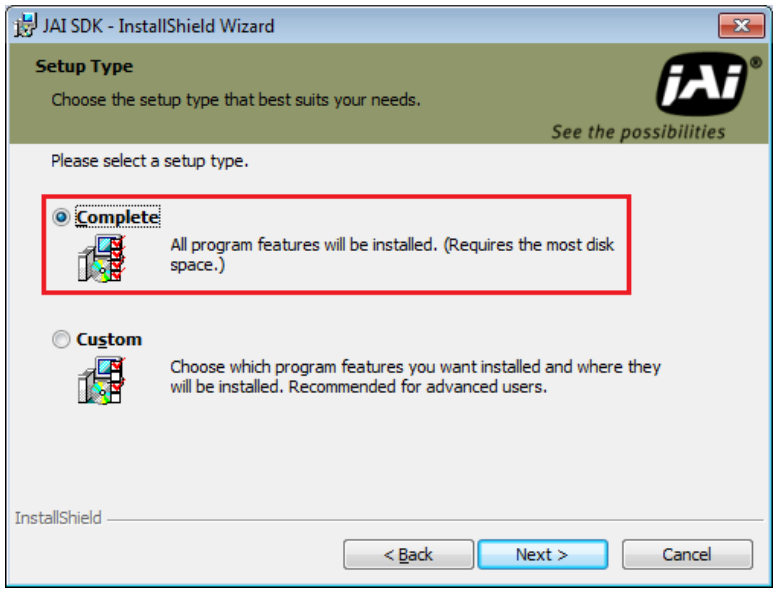
Figure 7. Destination Folder selection



3.5 Select Complete Setup

Normal setup installs the entire program; custom allows omission of portions of the install.

Figure 8. Setup type selection



3.6 Custom Setup

The user can choose not to install some portions of the JAI Control Tool Software. If changes to the installation are configured, click the Next button. If no alterations are made click the Back button and use the “Complete” option.

Figure 9. Custom setup type selected

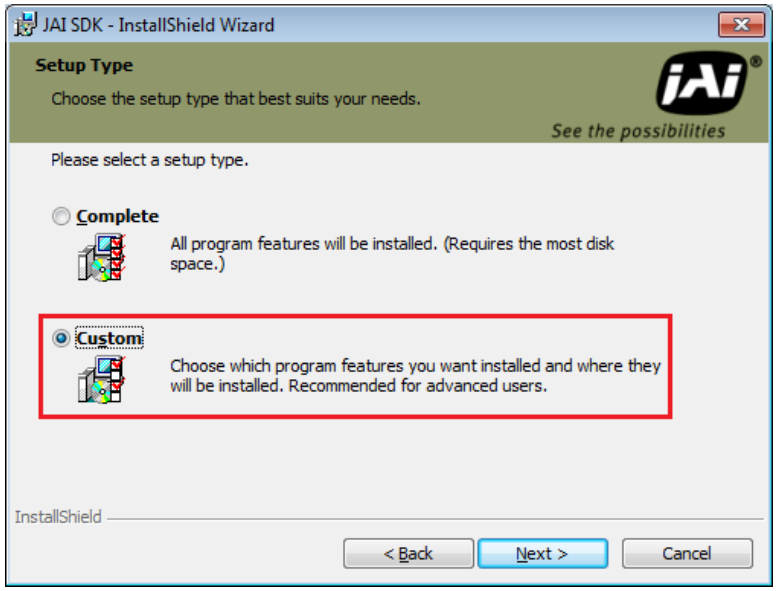
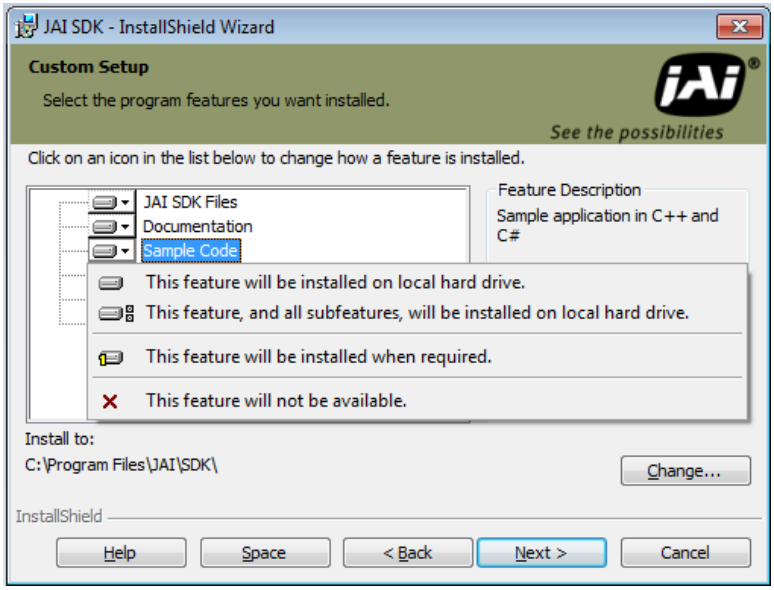


Figure 10. Custom feature selection



3.7 Filter Driver Installation Warnings

Depending on which Windows Operating System the JAI SDK is being installed on, different warnings display during the filter driver installation.

3.7.1 Security Warning on Windows Vista and Windows 7

The Windows Security driver software installation warning appears before Windows Vista or Windows 7 will install the filter driver. If you check the "Always trust software from 'JAI Inc.'" box, future installations will not ask for permission to install JAI, Inc. software.

Figure 11. Windows Vista and Windows 7 driver installation warning

3.7.2 Windows Logo Testing warning on Windows XP

The Windows Logo testing warning appears before Windows XP will install the filter driver. Continue with the installation by clicking on the Continue Anyway button.

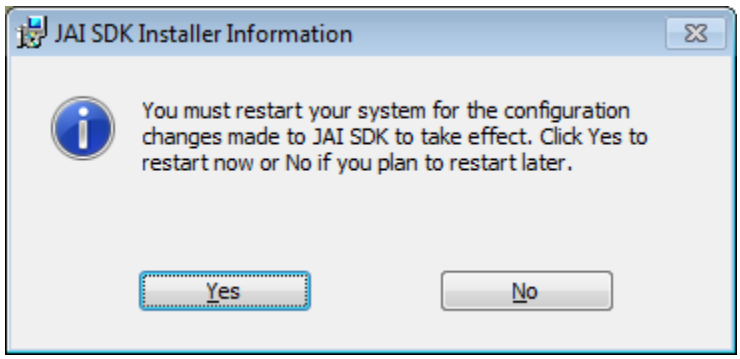
Note: The Windows Logo Warning serves two purposes. It asks if you want to install the device driver, and it warns that the driver has not passed Windows Logo testing to verify it is compatible with Windows XP.

Figure 12. Windows XP driver installation warning

3.8 Reboot

When the installer finishes, reboot the computer.

Figure 13. Reboot dialog on Windows Vista and Windows 7



4 Optimizing your computer for using the JAI SDK & Control Tool

4.1 Assigning IP-address to the network adapter in the PC

When no DHCP or no static IP is available for the NIC where cameras are connected, a Link-local address is automatically generated for the NIC when the camera is connected to the adapter. LLA can be a time consuming process and can take up to 120 seconds before the network is ready for communication with the cameras. The camera itself is ready to be used approximately 30 seconds after being connected to the computer. Therefore it is suggested to use DHCP or static IP address if possible.

Note: It is possible to assign a static IP address to the NIC that falls within the LLA address range. This will have the benefit that the cameras will normally boot right up on the correct subnet and therefore "Force IP" will be avoided.

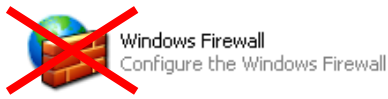
The LLA address range is IP address 169.254.xxx.yyy with a subnet mask of 255.255.0.0

4.2 Network interface cards

The JAI SDK & Control Tool makes use of a Filter Driver that is designed to work with all Gigabit Ethernet (IEEE 802.3) network cards.

As there can be differences in behavior of network cards, JAI recommends using Intel PRO/1000 series cards for optimal system stability

4.3 Firewalls



To ensure proper operation of the JAI SDK & Control Tool all firewalls must be disabled. This also includes the Windows firewall.

Click [Start], [Control Panel] for accessing the Windows firewall configuration.

Note: If all the cameras are GigE Vision® 1.1 devices (or newer) then they might have the feature called "Firewall Traversal" implemented. If this is the case then the firewall doesn't have to be switched off.

4.4 Setting Receive Buffers/Descriptors

If the Network Connection Properties list contains a property called Receive Descriptors/Buffers, then change this property to its maximum value supported by the NIC installed in your computer.

Click "OK" to save properties.

The standard default value is 256 bytes. The typical maximum which can be selected is 2048 bytes.

4.5 Setting Interrupt Moderation

If the Network Connection Properties list contains a property called Interrupt Moderation, then change this property to "Extreme" if it is available or to "On".

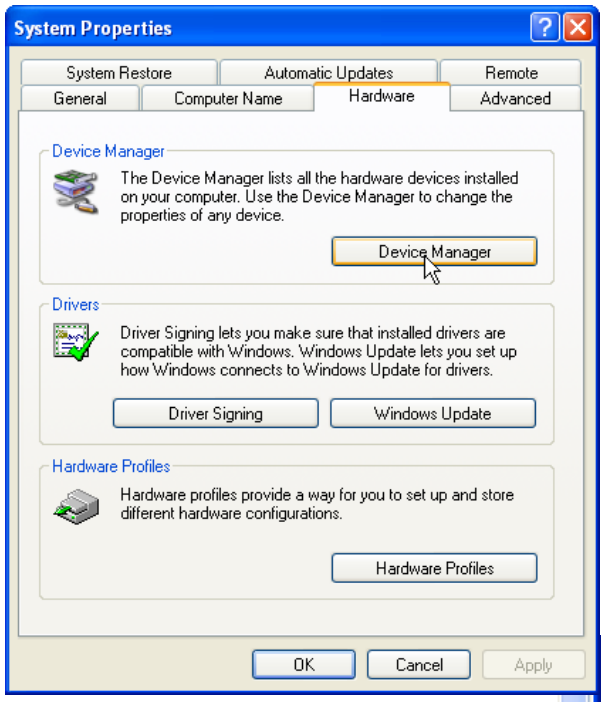
Click "OK" to save properties.

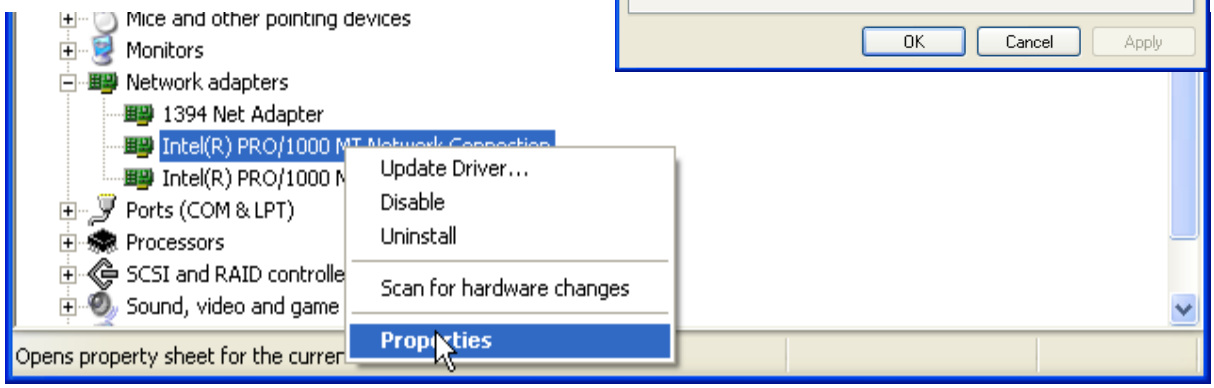
4.6 Enabling Jumbo Frames

The JAI GigE Vision® cameras can stream images using Jumbo Frames (sometimes also referred to as Jumbo Packets). By using Jumbo Frames, the overhead of the streaming is reduced, thus increasing the throughput.

Follow these instructions to enable Jumbo Frames:

1. Click [**Start**] and click [**Control Panel**].
2. Click [**Performance and Maintenance**].
3. Click [**System**].
4. Click [**Hardware**] tab.
5. Click [**Device Manager**].
6. Expand [**Network adapters**].
7. Select target NIC, **right-click**, and click [**Properties**].

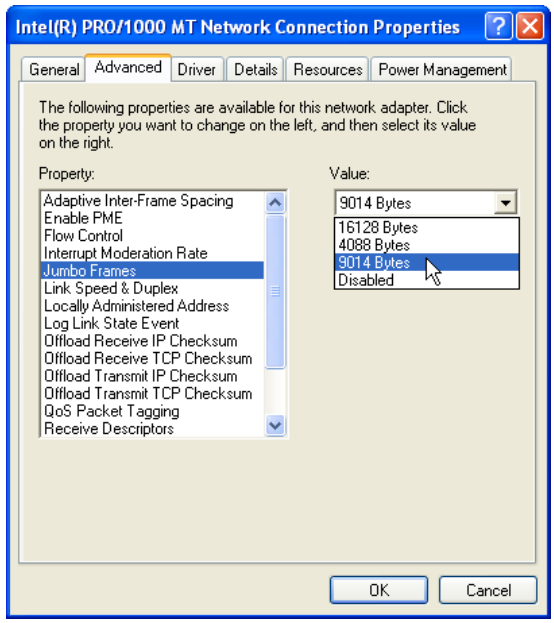




Note: The following procedure shows the setting for Intel PRO/1000 series.

The settings of Network Interface Cards from other manufacturers may show slightly different properties. If in doubt, contact the NIC manufacturer.

1. Click [Advanced] tab.
2. Select Jumbo Frames of Property, and select 9014 Bytes.
3. Click [OK].
4. Close [Device Manager].
5. Close [System Properties] by clicking [OK].



Note: When Jumbo Frames has been enabled for the NIC, it becomes possible to select a packet size larger than 1500 bytes inside the camera settings.

*The selected packet size in the cameras **must be** smaller than the maximum value selected for the NIC.*

4.7 Calculating and setting inter-packet delay

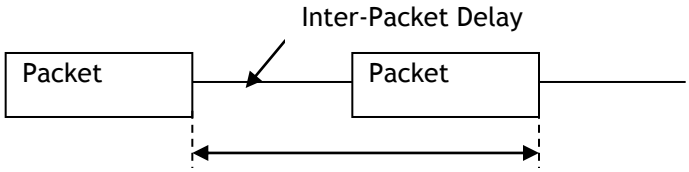
The GigE Vision® Stream Protocol automatically divides all images into smaller packets before transmission of the images and it provides a way to add delay in between these packets (Inter-Packet Delay). The Packet-Size can be adjusted using the JAI Control Tool.

When connecting several cameras to one network interface card via a switching hub, it is important to optimize the Inter-Packet Delay of the cameras to avoid congestion in the switch if one or more of the cameras transmit images at the same time. A typical switching hub is only capable of buffering a few image packets – not the whole image.

A sure sign of congestion is the loss of packets.

Since increasing the inter-packet delay also adds overhead to the data transfer it is important to calculate the optimal setting in order to make best use of the video bandwidth.

Figure 14. Duration of entire packet, including Inter-Packet delay



The JAI Control Tool has a built in wizard for calculating Inter-Packet Delay. The aim of the calculation is to "stretch" the image communication over time while still being able to achieve the specified frame rate. This will leave "gaps" between each image packet that can be used for transferring image packets from other cameras. Furthermore it is important to mention, that the gaps between each packet must be larger than the packages coming from other cameras. Practically, this means that the packet size for all cameras must be adjusted to fit the Inter-Packet Delay which has been set. Otherwise, it would not be possible to merge the communication.

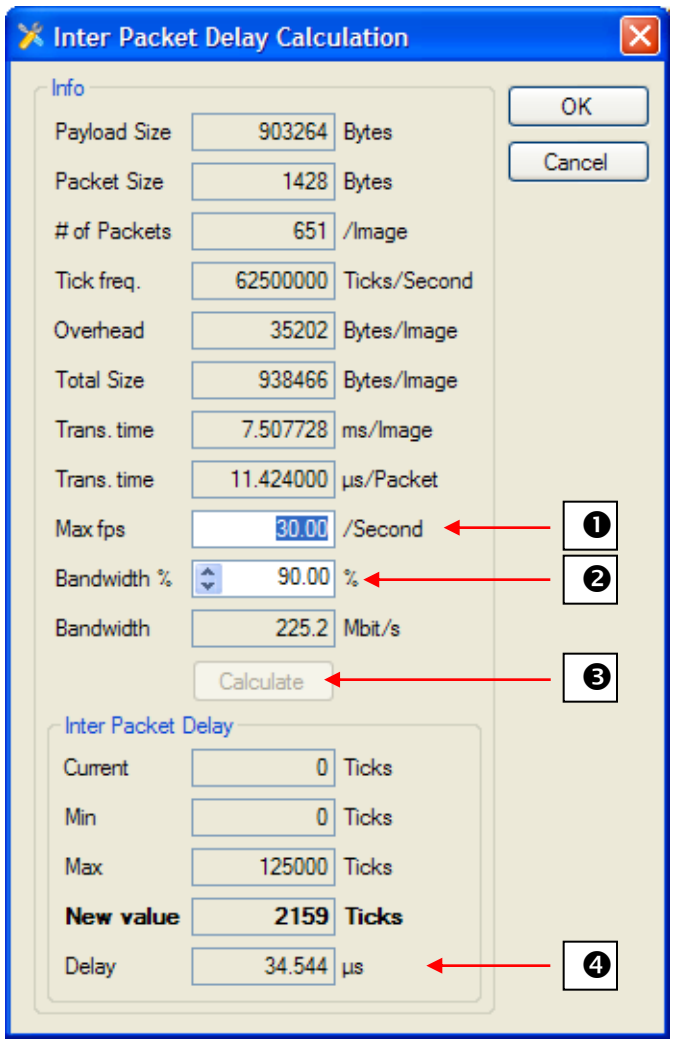
If either the Packet-Size or the Pixel-Format later on is changed for a camera then the Inter-Packet delay needs to be re-calculated using the Inter-Packet Delay wizard.

When the Inter-Packet Delay function is selected, a button appears on the right hand side of the bar.

Click the button to open the Inter-Packet Delay calculation wizard window.

Figure 15. Opening the Inter Packet Delay Calculation wizard

Figure 16. Inter Packet Delay Calculation wizard



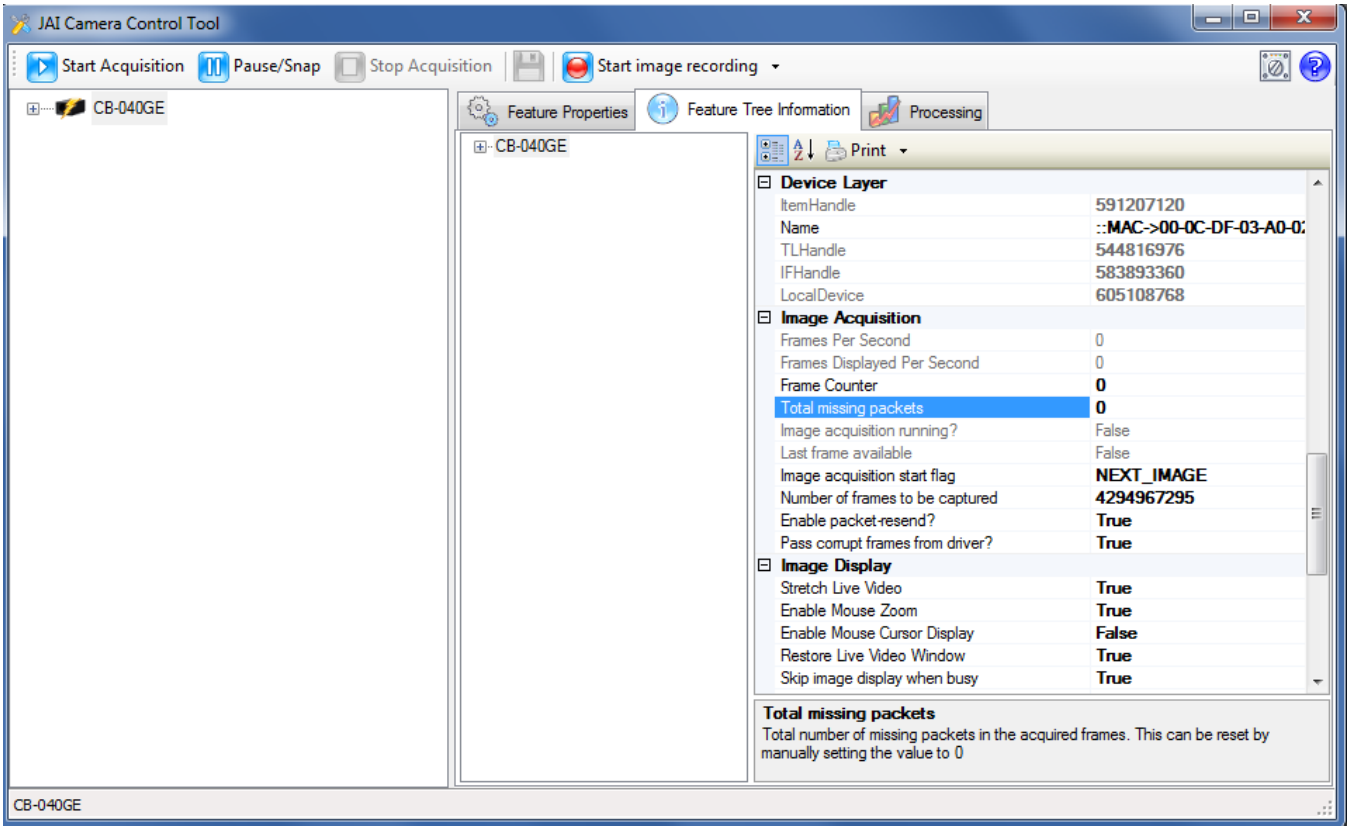
1. At first, type in the maximum needed frame rate of the selected camera.
2. Set the expected bandwidth usage at 90%-95% to leave some room for re-transmissions.
3. Click the Calculate button.
4. New value is calculated.
5. Click OK. This shown value is automatically transferred to the Packet Delay column of the Control Tool

In the Inter-Packet Delay Calculation dialog the transmission time for each package is displayed. Additionally, the new Inter-Packet Delay is displayed in microseconds which allow users to verify whether or not the delay is larger than individual packets from the other cameras.

4.8 Further hints for optimizing network Settings

The number of packages lost is directly displayed in the Control Tool under “Feature Tree Information”. This value is the amount of all packages lost and can be reset manually.

Figure 17. Display of the total missing packets



If after having followed the above steps for optimizing the transmission you still experience lost packets (the symptom of lost packets is horizontal black lines in the captured image), you should try these additional settings:

4.8.1 Turning Hyper Threading off

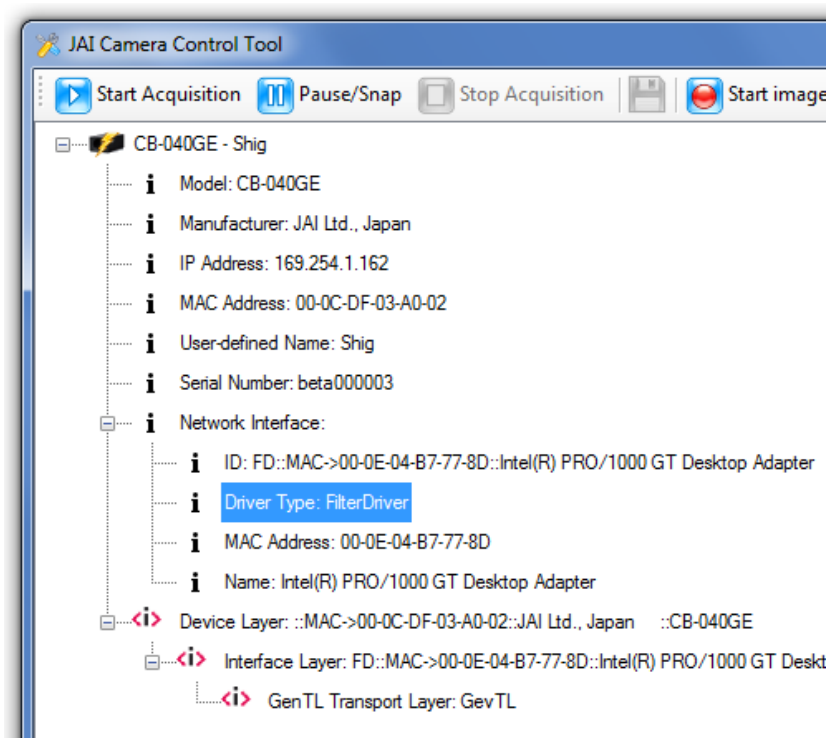
This is done in BIOS settings. The method for changing this setting depends on the computer. Please refer to the computer/motherboard manufacturer's reference manual.

4.9 Selecting Filter Driver or Windows socket driver

The Filter Driver is part of the JAI SDK & Control Tool installation.

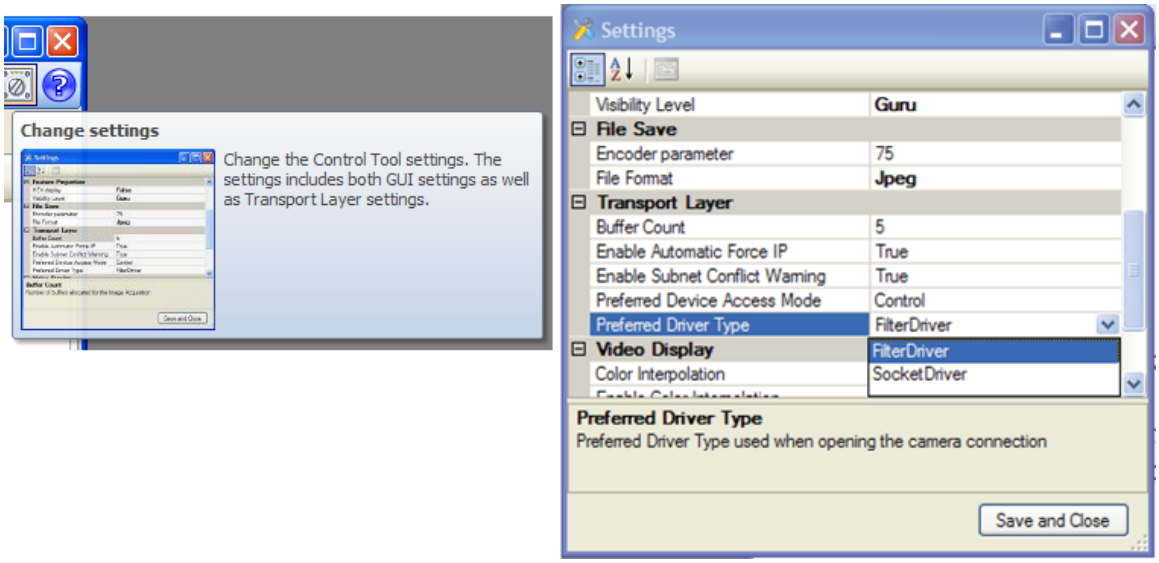
To confirm the Filter Driver installation, after the camera is connected, the JAI Camera Control tool shows the Filter Driver as the network Interface.

Figure 18. Driver type display



If you wish to use the Widows Socket Driver, click the settings button and open the setting menu and make your selection.

Figure 19. Selecting preferred driver type in the Settings dialog



5 Connecting to GenTL™, Camera Link®, and USB3 Vision™ Devices

5.1 Connecting to GenTL™ Providers

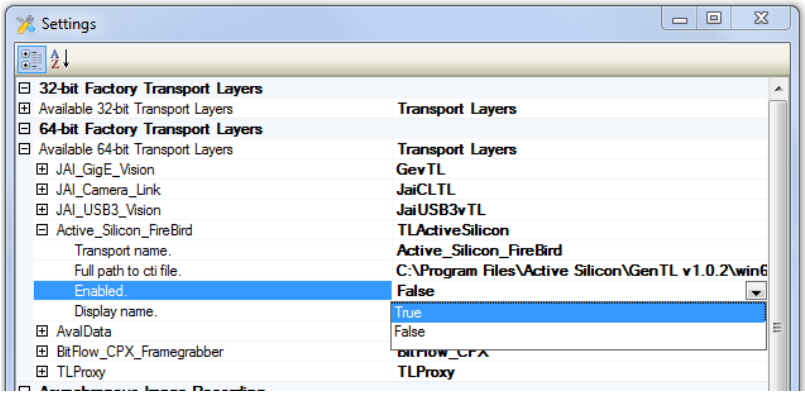
The control tool makes it very easy to evaluate GenTL™ compliant devices because it is a GenTL™ compliant consumer (i.e. a "plug-and-play") graphical user interface, where all GenTL™ compliant cameras will be automatically detected, and it is easy to connect to any of these cameras and both display and control all available camera features. GenTL™ interfaces includes, among others, GigE Vision®, CoaXPress®, Camera Link® (supporting the GenCP protocol), USB3 Vision™, etc.

Note that this also applies to controlling cameras through the JAI SDK: once configured using the control tool, the aforementioned camera interfaces may be controlled using the functions in the JAI SDK. In most instances, the same code may be used to control all of the camera interfaces. (Exceptions include CoaXPress®, where both the camera *and* the framegrabber must be controlled and synchronized using JAI SDK functions. Other exceptions may exist). All camera interfaces (**except** for Camera Link®) support streaming in the control tool or via the SDK. In addition, JAI Camera Link® cameras that support control via an ASCII command set continue to do so.

5.1.1 Enabling GenTL™ Producers

GenTL™ producer DLLs must be registered with the system. Just click the "Change Settings" button on the upper right-hand corner of the Control Tool and edit the list of registered producers:

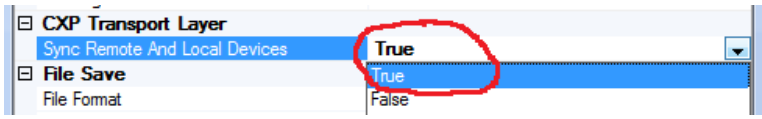
Figure 20. GenTL™ producer properties in the Settings Dialog



Make sure to enable all desired transport layers, and ensure that the "Full path to cti file" points to the correct path. Two JAI producers (for GigE Vision® and USB3 Vision™) are enabled by default.

5.1.2 Connecting to CoaXPress (CXP) Framegrabbers

A new property has been added to the Settings Dialog which enables the synchronization of width/height values between cameras and CXP framegrabbers. The property, "Sync Remote and Local Devices" must be enabled (i.e. set to True) manually by the user. When enabled, changes made to height/width/pixel-format in the camera will be set in the framegrabber (and vice-versa).



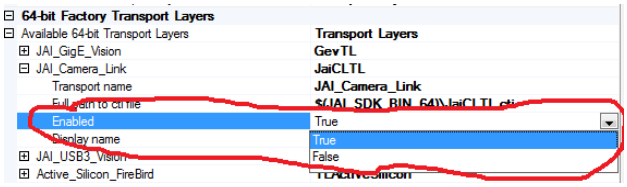
5.2 Connecting to the JAI GenCP-Enabled Camera Link® Provider

JAI GenCP-Enabled Camera Link® cameras may also be controlled in a manner similar to GigE® cameras. However in order to view Camera Link® compliant cameras in the control tool, the JAI camera link transport layer "JaiCLTL" must be enabled (see section 5.1.1 for more details).

5.2.1 Configuring Automatic Discovery of JAI GenCP-Enabled Camera Link® Cameras

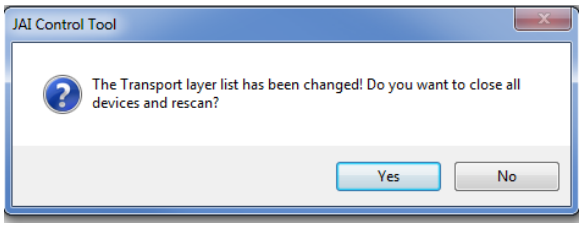
Camera Link® discovery is disabled by default. To enable Camera Link® discovery, set the "Enabled" entry to "True" for the "JAI_Camera_Link" transport layer (32-bit and/or 64-bit) in the Settings dialog:

Figure 21. Configure Camera Link® Transport Layer (64-bit shown)



After pressing the "Save and Close" button, the following dialog will appear:

Figure 22. Rescan Devices Dialog

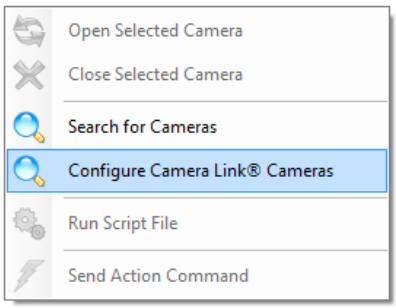


Select "Yes" to rescan for new cameras. If a camera link camera is attached to a framegrabber, it should appear in the camera tree-view pane.

5.2.2 Manually Configuring JAI GenCP-Enabled Camera Link® Cameras

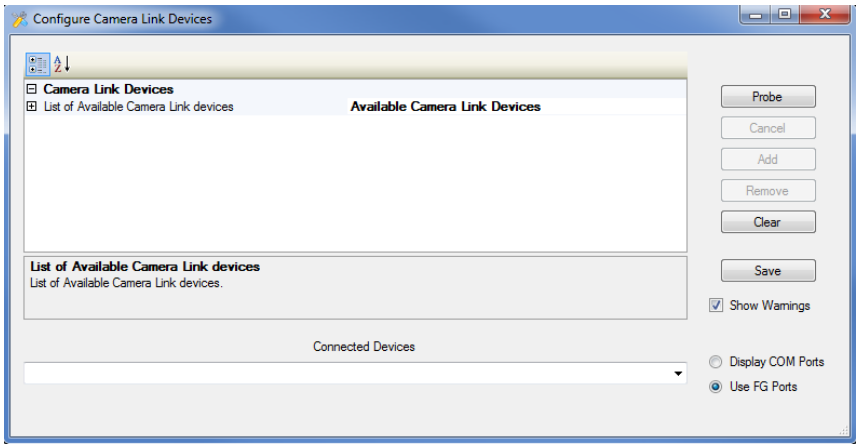
JAI GenCP-Enabled Camera Link® cameras may be discovered manually by "right-clicking" the camera tree-view and select "Configure Camera link Devices" in the pop-up context menu.

Figure 23. Configure Camera Link® Cameras popup menu



The following dialog box will appear:

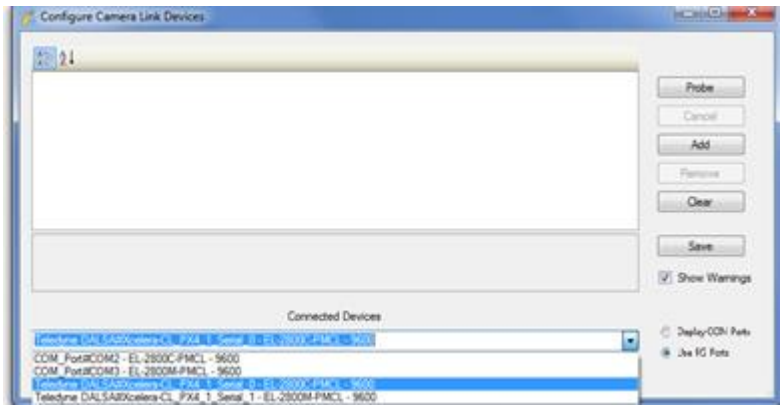
Figure 24. Configure Camera Link® Cameras Dialog



Press the "Probe" button to discover cameras currently active and connected to a serial port (i.e. a framegrabber's serial port).

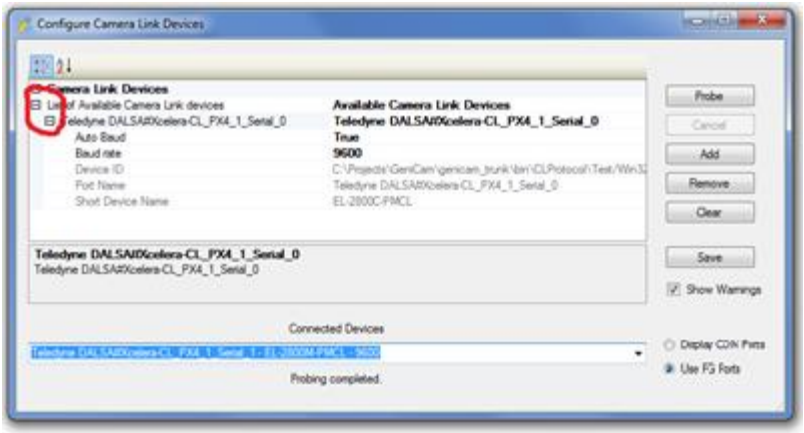
Any devices discovered during this process will be listed in the "Connected Devices" combo box.

Figure 25. Selecting a camera in the Configure Camera Link® Cameras Dialog



The user may add cameras to the configuration file by selecting a camera of interest in the combo box and pressing the "Add" button (details may be viewed by clicking on the "(+)" buttons). Pressing the "Add" button will move items from the "Connected Devices" combo box to the list of available Camera Link® devices. Press the "Save" button when done.

Figure 26. Camera properties in the Configure Camera Link® Cameras Dialog

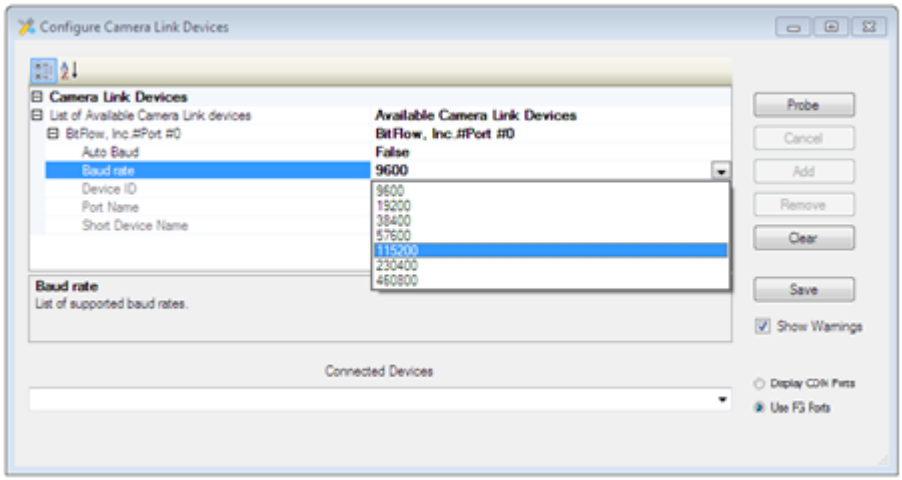


Note that only the following entries should be modified:

1. Auto Baud – Set to true if the camera's XML file will be downloaded using the highest baud rate supported by the camera.

2. Baud Rate – The baud rate at which to connect to the camera for normal operation. The user may select from any of the baud rates supported by the camera:

Figure 27. Baud rate properties in the Configure Camera Link® Cameras Dialog



The next time a "Search for Cameras" performed, only the cameras saved in the configuration file will be listed in the "camera tree-view". Probing will not be necessary.

5.3 Connecting to USB3 Vision™ Cameras

USB3 Vision™ cameras are an emerging player in machine vision. However, the standard is evolving, and not all devices meet the USB3 Vision™ standard. In particular, USB 3.0 controllers (and their drivers) may not operate correctly with USB3 Vision™ cameras, even though the cameras themselves meet the USB3 Vision™ specification. It is strongly recommend that the manufacturer of your USB 3.0 controller be verified as supporting USB3 Vision™ cameras.

Note: USB 3.0 cameras are not supported on Windows XP and earlier operating systems.

5.3.1 Configuring USB3 Vision™ Cameras

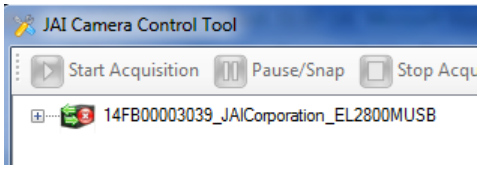
USB3 Vision™ cameras must be configured to use the JAI USB 3.0 driver before they can control by the Control Tool.

1. Connect a USB3 Vision™ camera to the host computer.
    a. For JAI USB3 Vision™ cameras:
        i. Connect the power cable
        ii. Wait for the "POWER/Trig" LED to turn green
        iii. Plug in the USB cable.
2. The camera MUST appear in the Windows Device Manager (see section 4.6 "Enabling Jumbo Frames") as shown below:

Figure 28. Device Manager display of USB3 Vision™ device with no driver attached

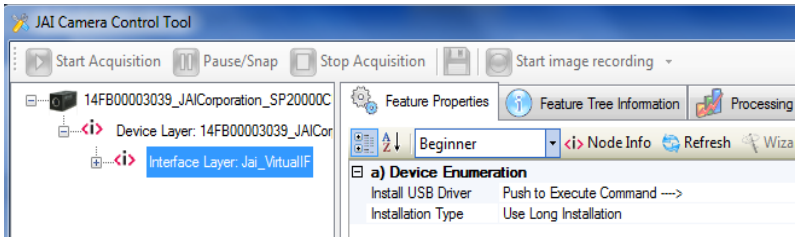
3. Open the Control Tool (or right-click in TreeView if the application is already running) and select "Search for Cameras". An entry for the camera should appear as an item in the tree (the actual text will vary):

Figure 29. Initial display of a USB3 Vision™ camera in the Control Tool



4. Select the camera with the mouse. If the icon remains red, the JAI USB 3.0 driver must be loaded.
5. Select the "+" icons until the "Interface Layer" is displayed:

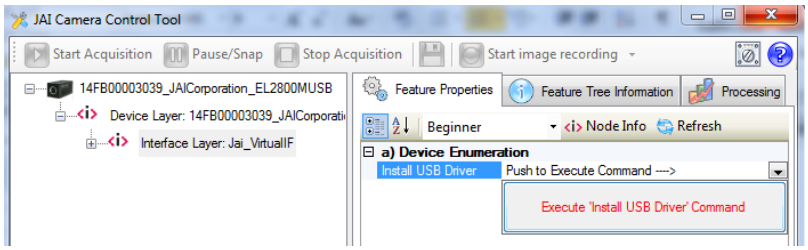
Figure 30. The "Interface Layer" of the USB3 Vision™ camera without the JAI USB 3.0 driver



6. Select the Installation Type – The user may select between three installation types:
    a. Long installation – Does not require a system reboot (if the Control Tool is run with administrator privileges). Does not require the Control Tool to be restarted.
        i. After the driver has been installed, the user will be prompted to unplug the USB cable from the camera and plug it in again so that the control tool application does not have to be restarted to use the camera.
        ii. This should only be necessary the very first time the driver is installed.
        iii. After installation the user will be prompted to optionally switch to the short installation for subsequent cameras.
    b. Short installation – The computer may need to be rebooted after installing the driver the very first time.
        i. No prompts will be displayed after the driver has been installed.
    c. Always use the Long Installation – Will not ask the user to optionally switch to the short installation for subsequent cameras.

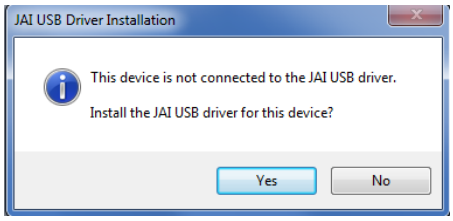
7. Press "Push to Execute Command --->" button next to the "Install USB Driver" property.

Figure 31. "Install USB Driver" property



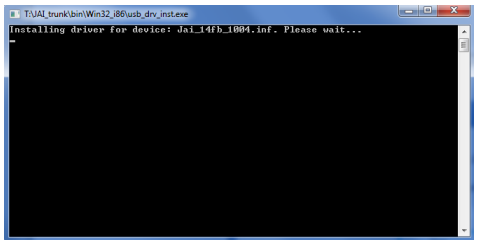
8. Press the "Execute 'Install USB Driver' Command" button. The following dialog will appear. Press "Yes" to continue.

Figure 32. USB Driver installation dialog: "This device is not connected…"



9. A command window will appear indicating that the driver is being installed for the selected camera:

Figure 33. USB Driver installation command window

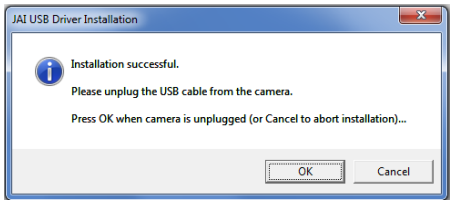


10. The command should remain visible for ~10-15 seconds (i.e. is should not disappear immediately).
11. If the installation is successful, the following dialogs will appear (Long Installation only. Skip to step 15 for the short installation).

    The Control Tool can now be closed and reopened, and the camera should be available for control and image acquisition.

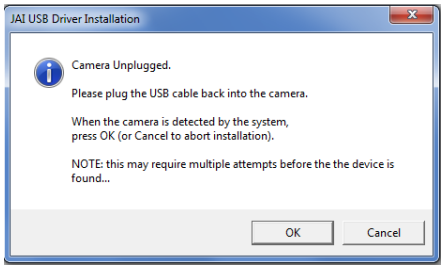
    In order for the device to be available in the Control Tool without have to close and reopen it, the camera must first be disconnected and then re-connected from/to the host computer. The following steps guide the user in this process. The first dialog requests that the user unplug the USB cable:

Figure 34. USB Driver installation dialog (Long Installation): "Please unplug the USB cable…"



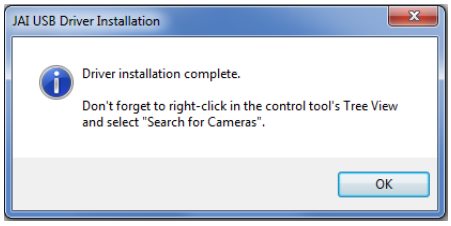
12. Press OK after the cable has been unplugged (another dialog will alert the user if OK is pressed without unplugging the cable). If application has detected that the cable has been unplugged, the following dialog will appear:

Figure 35. USB Driver installation dialog: "Please plug the USB cable back…"



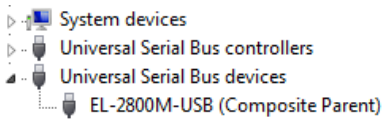
13. Plug the USB cable back in, then press OK. The application will attempt to detect when the host computer has loaded the driver for the camera.
14. **NOTE: this process may take a few seconds.** If the OK button before the driver is loaded, a dialog will appear allowing the user to wait and "try again".
15. When the installation is complete, make sure to right-click in the Control Tool's tree view and select "Search for Cameras".

Figure 36. USB Driver installation dialog: "Driver installation complete."

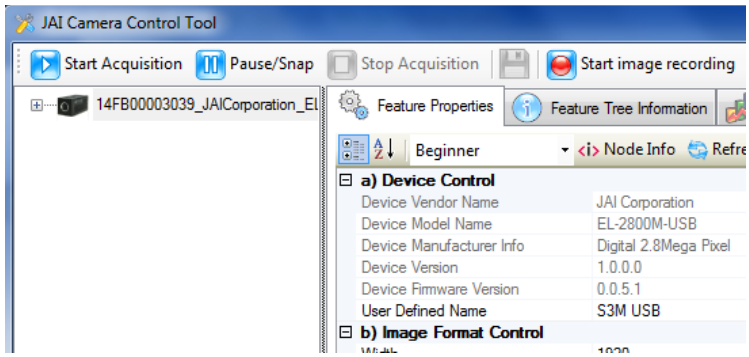


16. The device will appear in Device Manager below "Universal Serial Bus controllers", similar to that shown below (again, the actual text will vary):

Figure 37. Device Manager showing a USB3 Vision™ camera with the JAI USB 3.0 driver installed



17. In the Control Tool, right-click in the Tree View and select "Search for Cameras". When complete, select the camera icon again. If the driver was successfully loaded for the camera, the XML data should then appear.



6 General advice on using the Camera Control Tool

The examples below show the Control Tool together with a CB-040GE camera. When connecting to any given camera, the Control Tool will show the name and properties of that particular camera.

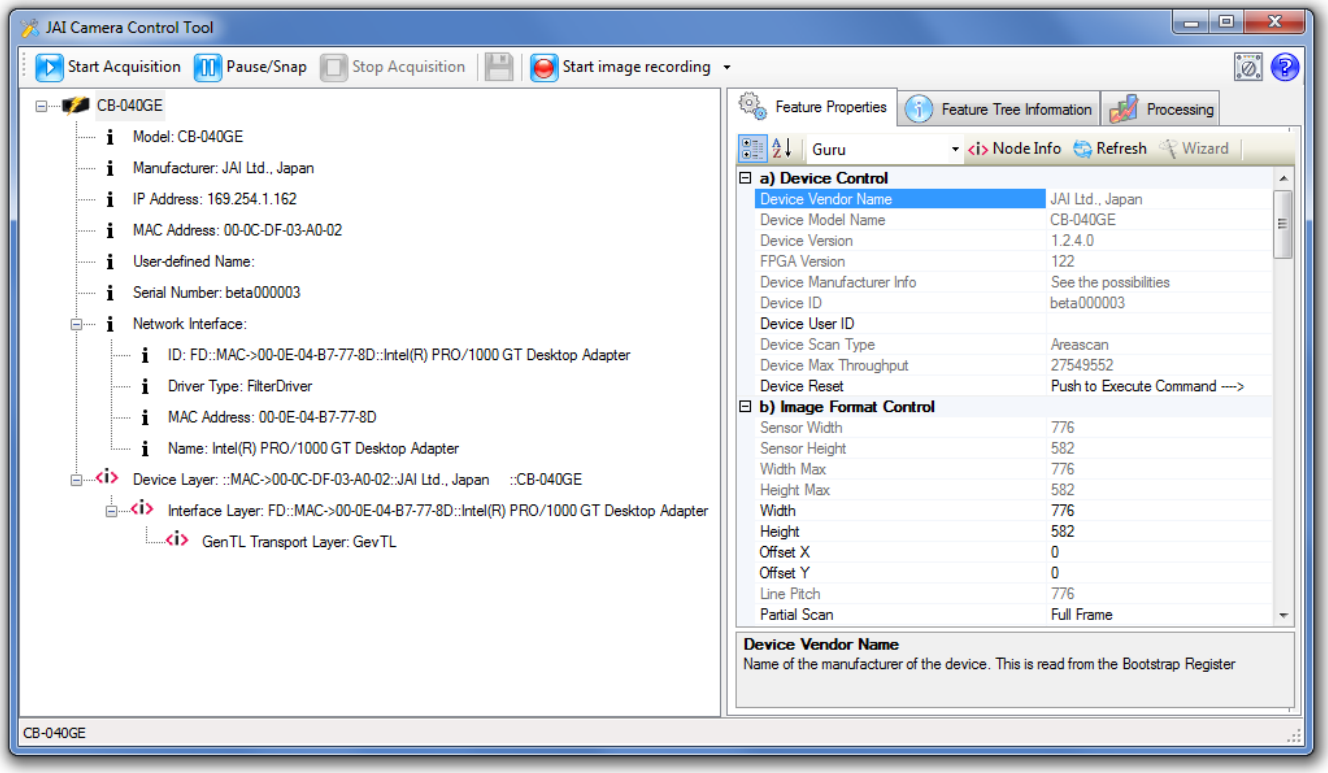
After starting the JAI Control Tool the below window will be shown. All connected cameras will automatically be detected using the GigE Vision® Device Discovery broadcast and shown in the Camera Tree-view on the left-hand side of the Control Tool.

The connection to the cameras is established by simply clicking the cameras in the Tree-view.

Please note that in this example Windows uses the LLA (Link Local Address) method to assign an IP address to the NIC adapter. This process can take up to a minute to complete and if the Control Tool is started before completion of the LLA then no cameras will be detected.

Once the cameras are detected and the connection icon shows connected status, the camera connection has been successfully opened. Now the camera properties (parameters) can be set and streaming can be initiated.

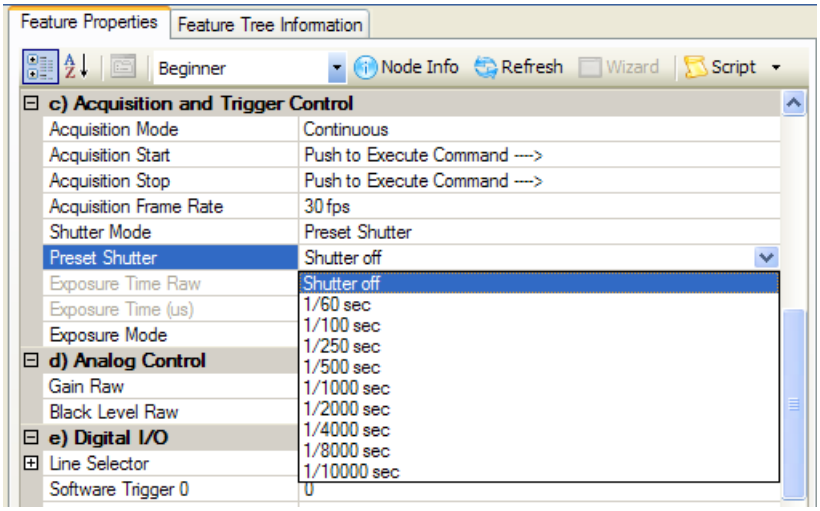
Figure 38. Display of all camera feature properties



Changing parameter settings is done either by selecting a value in a pull-down menu (as shown in the Present Shutter example below), moving a slider bar (as shown in the Image Format Control/Width example below) or typing the desired value directly into the text box.

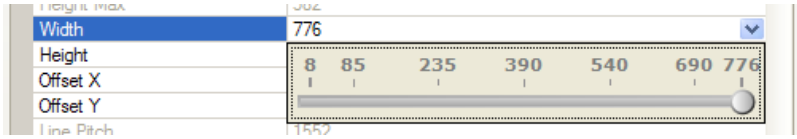
The available ways to change a parameter depends on the value type of the given parameter. If the feature value is an “enumerations” or a Boolean then it will be shown as a complete list of possible values in a pull-down menu.

Figure 39. Example of setting a parameter by selecting from a pull-down menu



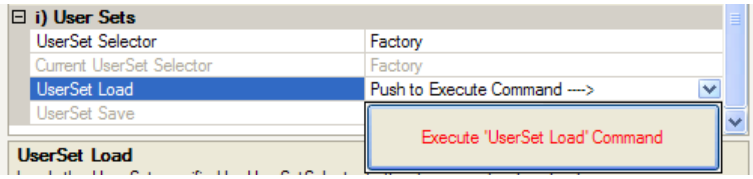
If the value is an "integer" or a "floating-point" value then the value will be displayed as a slider bar where the minimum, maximum and possibly an increment size is displayed.

Figure 40. Setting "Width" by using the slider bar and/or typing a value directly into the text box



If the value is a "command" then it will be controlled using a button (as shown in the "UserSet Load" sample below).

Figure 41. Executing the "UserSet Load" command



If the parameter is a text string then it is simply edited by selecting the text box and typing in the new value

Figure 42. Changing the "Device User ID" text string

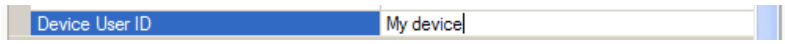
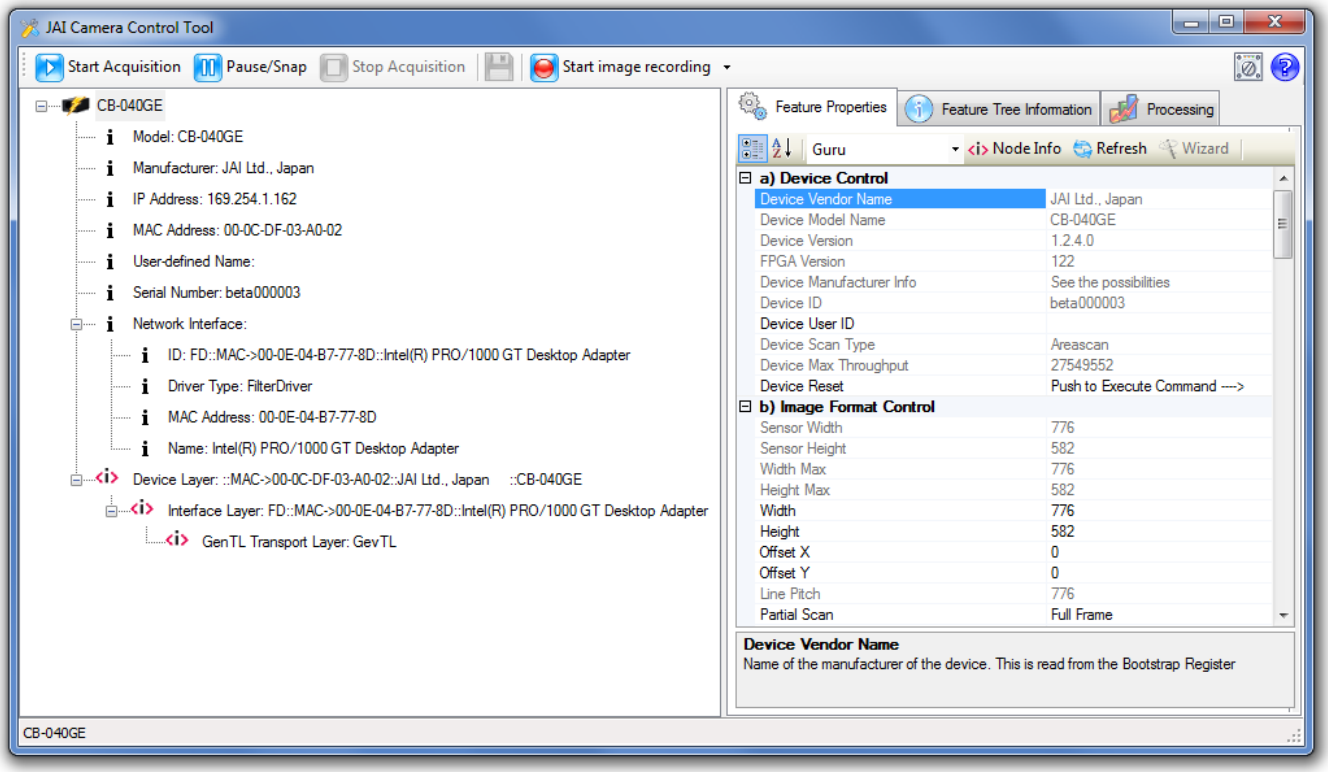


Figure 43. Acquisition is initiated by clicking the "Start Acquisition" button



During acquisition many parameters in the Feature Properties list are grayed out, which means that they cannot be changed during image acquisition.

It is possible to change parameters such as shutter and gain values during image acquisition.

When for instance the image size needs to be changed, the image capturing must be stopped by clicking “Stop Acquisition”.

For the further details, please refer “JAI Control Tool” document provided with JAI SDK.

7 Compatible programming languages and tools

The JAI SDK consists of two APIs (Application Programming Interfaces):

- Jai_Factory.dll: Windows DLL with "Plain" C-interface and C/C++ header file with all functions and parameter types defined. This makes it possible to use JAI SDK in almost any programming language available for the Windows platform.
- Jai_FactoryDotNET.dll: Managed Windows DLL (.NET Framework 2.0). This DLL exposes the "Plain" function interface PLUS an "object-oriented" API. The object-oriented API encapsulates the "standard use" of the "plain" C-interface API.

Reference documentation for the two APIs is available in both PDF- and CHM-format as part of the software download.

A number of sample applications are included with the software download. It demonstrates to the users how to use the API's (in both C++ and C# for Microsoft Visual Studio 2005).

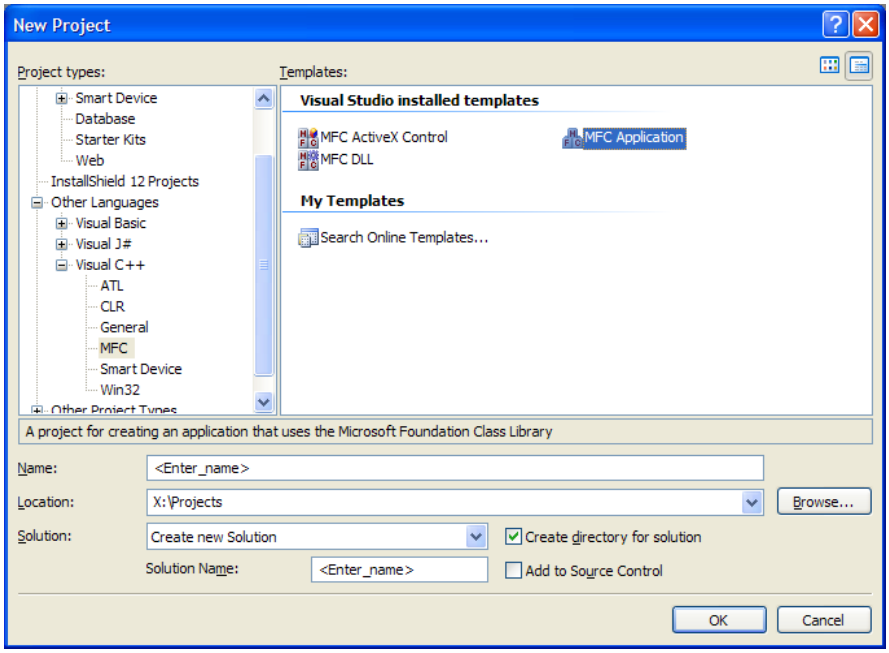
7.1 Using JAI SDK together with Microsoft Visual Studio 2005 C/C++

The JAI SDK comes with many sample project written in C++ using Microsoft Visual Studio 2005.

The following steps needs to be followed in order to use the Jai_Factory.dll.

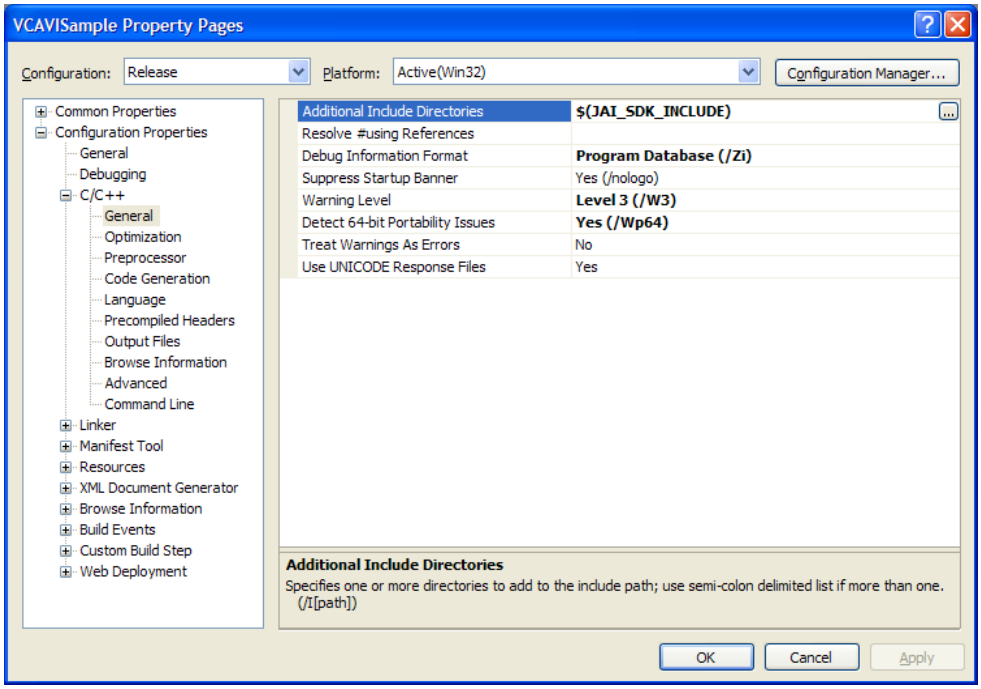
1. Create a new C++ project using the File->New->Project menu point.

Figure 44. New C++ MFC project in Visual Studio 2005



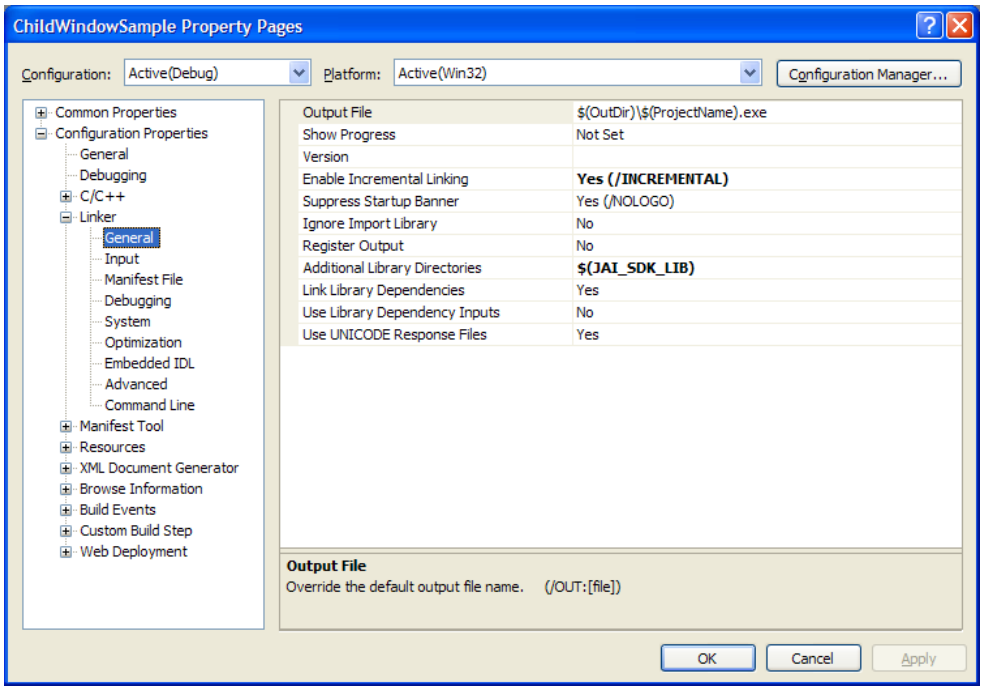
2. Add the Jai_Factory.h to the main project header file (typically the Dialog-class header file if it is a MFC-project). This makes it possible to add member variables of the JAI SDK specific types defined in the Jai_Factory.h (such as FACTORY_HANDLE and structures like J_tIMAGE_INFO)
3. Add the Environment Variable "$(JAI_SDK_INCLUDE) under the "Additional Include Directories" under the "C/C++->General".

Figure 45. Sample project property page from Visual Studio 2005



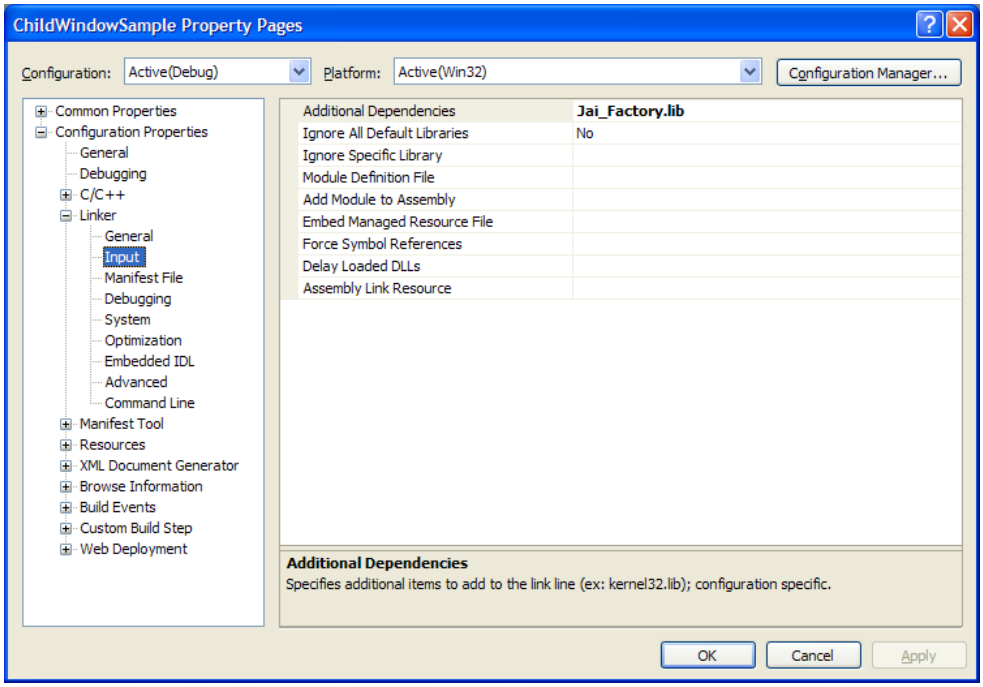
4. Add the Environment Variable "$(JAI_SDK_LIB)" to the "Additional Library Directories" under the "Linker->General" settings.

Figure 46. C++ Linker settings in Visual Studio 2005



5. Add the Jai_Factory.lib as "Additional Dependencies" under the "Linker->Input" settings

Figure 47. Adding the library file as additional dependency



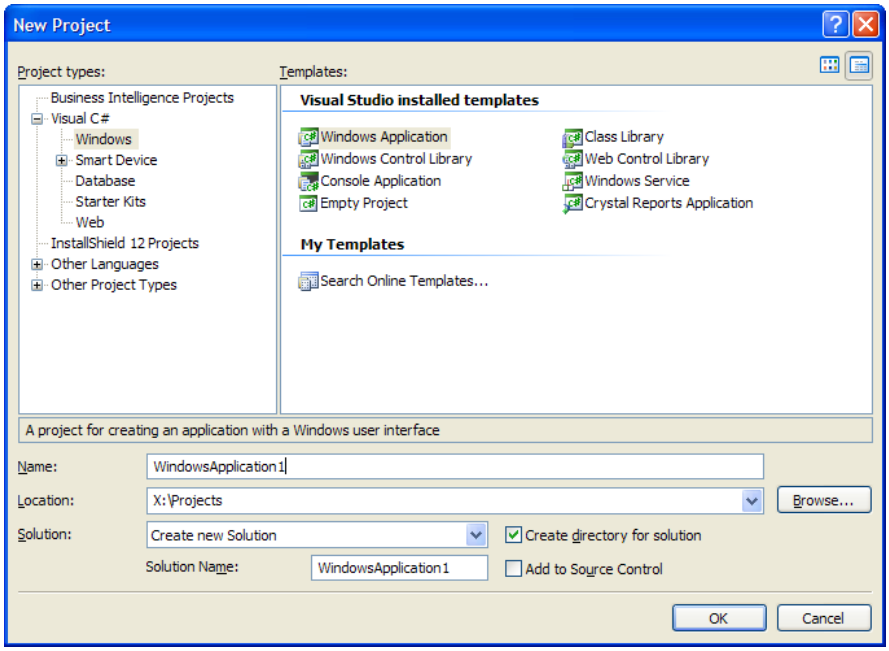
7.2 Using JAI SDK together with Microsoft Visual Studio 2005 C#

The JAI SDK comes with many sample project written in C# using Microsoft Visual Studio 2005.

The following steps needs to be followed in order to use the managed Jai_FactoryDotNET.dll.

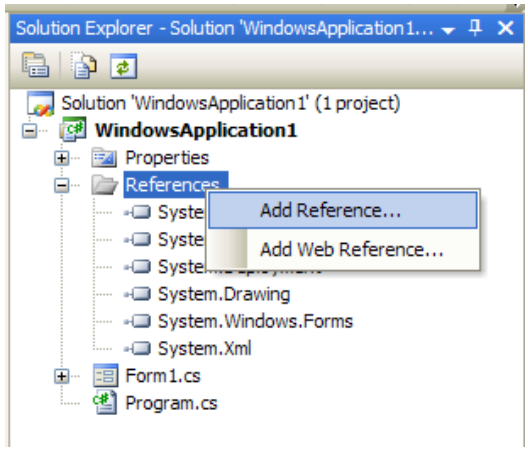
1. Create a new C# project using the File->New->Project menu point.

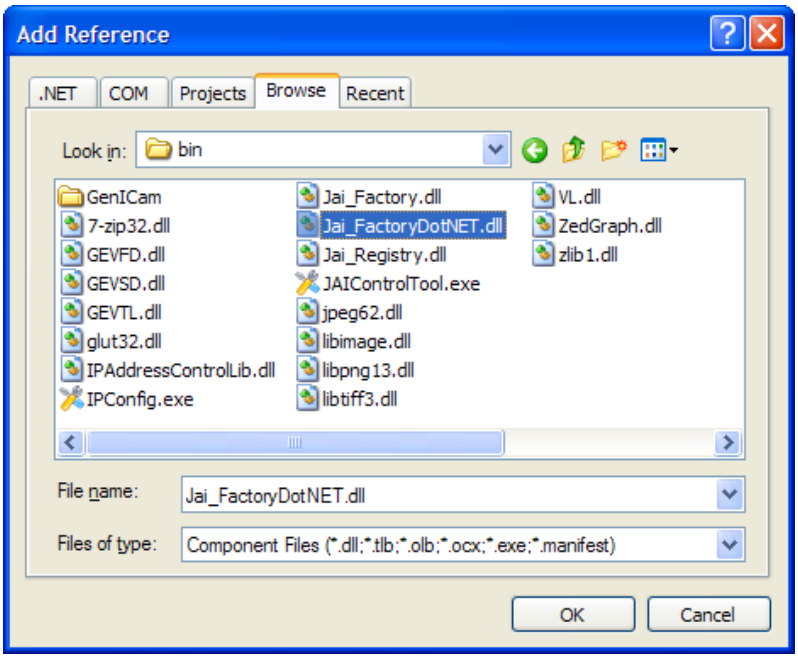
Figure 48. New C# project in Visual Studio 2005



2. Add the Jai_FactoryDotNET.dll from <Program Files Directory>\JAI\SDK\bin as a reference to the project.

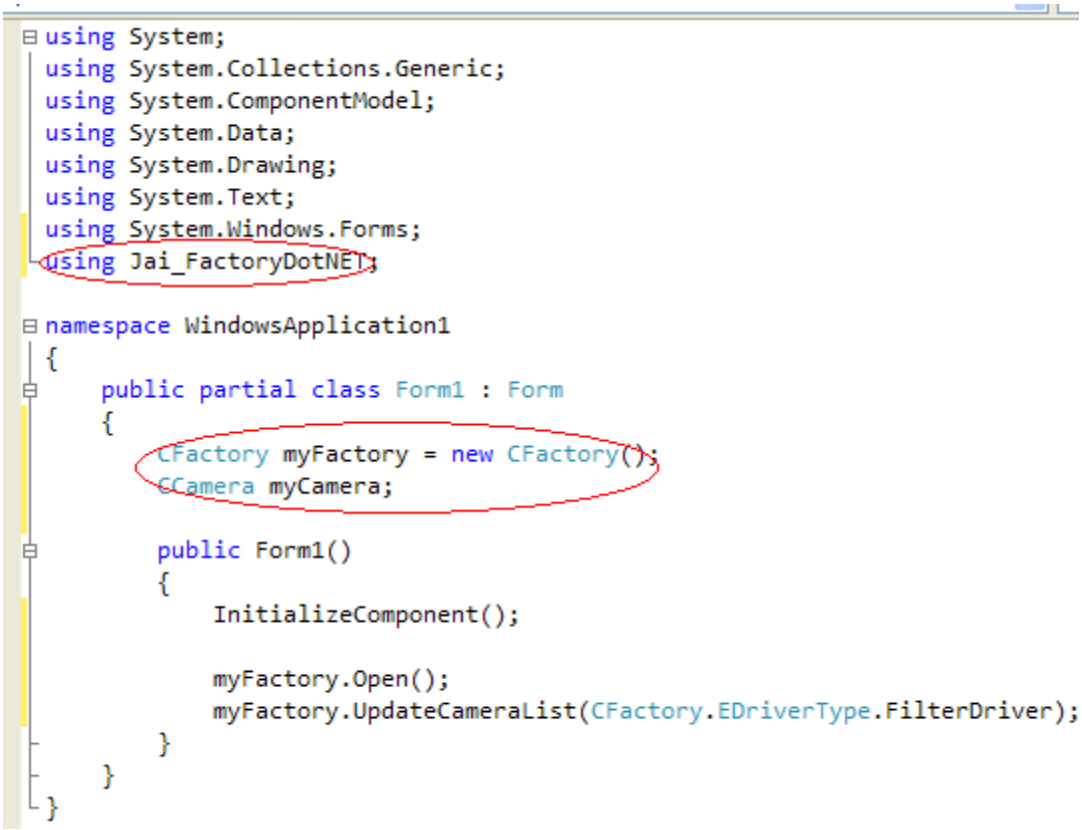
Figure 49. Adding Jai_FactoryDotNET.dll as a reference





3. Add the line “using Jai_FactoryDotNET;” to the main form source code and create and use the CFactory, CCamera and CNode classes like described in the JAI SDK documentation.

Figure 50. Use the Jai_FactoryDotNET in C# application



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Jai_FactoryDotNET;

namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        CFactory myFactory = new CFactory();
        CCamera myCamera;

        public Form1()
        {
            InitializeComponent();

            myFactory.Open();
            myFactory.UpdateCameraList(CFactory.EDriverType.FilterDriver);
        }
    }
}
```

7.3 Using JAI SDK together with Microsoft Visual C++ 6.0

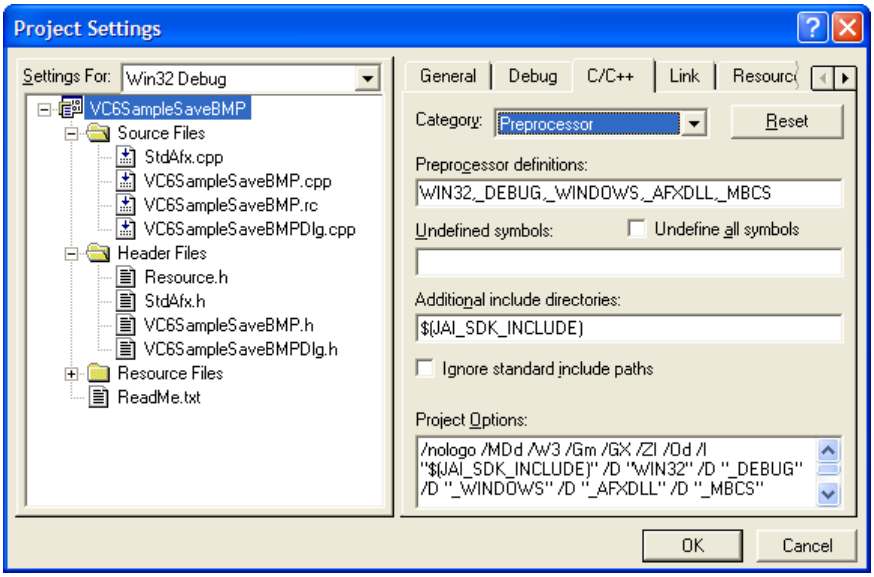
Even though the JAI SDK only comes with sample project written using Microsoft Visual Studio 2005 it is still possible to use the Jai_Factory.dll in Microsoft Visual C++ 6.0 project.

The following steps needs to be followed in order to use the Jai_Factory.dll.

1. Create a new Visual C++ 6.0 project using the File->New menu point.
2. Add the Jai_Factory.h to the main project header file (typically the Dialog-class header file if it is a MFC-project). This makes it possible to add member variables of the JAI SDK specific types defined in the Jai_Factory.h (such as FACTORY_HANDLE and structures like J_tIMAGE_INFO)

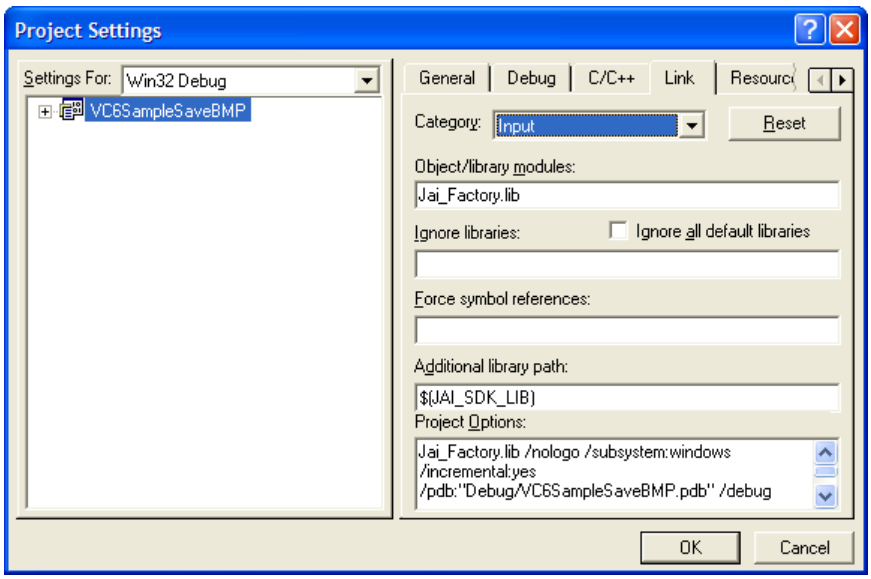
3.In the Project Settings the Environment variable “$(JAI_SDK_INCLUDE)” needs to be added as “Additional Include Directory”. This is located on the “C/C++ tab” under the “Preprocessor” category.

Figure 51. C/C++ project settings needed in Visual C/C++ 6.0



Jai_Factory.lib needs to be referenced in the "Link tab" in order to automatically load and link to the Jai_Factory.dll. In the "Input" Category the Jai_Factory.lib is referenced as well as the Environment Variable "$(JAI_SDK_INCLUDE)".

Figure 52. Linker project settings needed in Visual C/C++ 6.0



7.4 Using JAI SDK together with Microsoft Visual Studio 2008 and Visual Studio 2010:

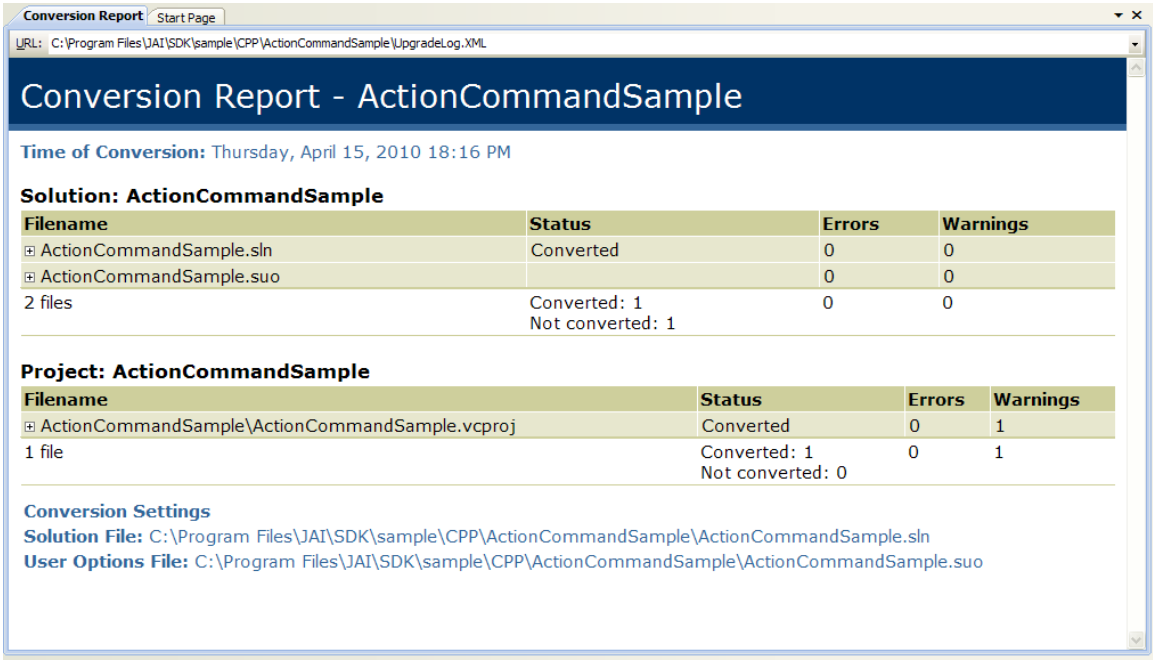
All sample application are currently created using Microsoft Visual Studio 2005 but it is indeed possible to use the sample applications in Microsoft Visual Studio 2008 and Visual Studio 2010 as well! If an existing sample is opened using Microsoft Visual Studio 2008 or Visual Studio 2010 then the user will automatically be asked if the project should be converted to the newer version of Visual Studio.

Figure 53. Visual Studio Conversion Wizard.

Simply follow the steps required by the wizard and at the end you will be able to get a Conversion Report for the conversion process:

Figure 54. Conversion Report

Conversion Report from converting the C++ `ActionCommandSample` application.



7.5 Dynamic load of `Jai_Factory.dll`

Sometimes it is desirable to postpone the loading of the `Jai_Factory.dll` until it is actually needed by the user application. This will for instance make it possible to startup the user application without having the JAI SDK installed on a system in case the JAI SDK functions are not currently needed.

JAI SDK now includes a new sample application that illustrates how this can be done. The sample is a C/C++ sample and it is called `DynamicLoadSample`.

The basic idea is to include a new header file named `Jai_Factory_Dynamic.h` inside one of the source files for the user application so that the code inside `Jai_Factory_Dynamic.h` will be embedded and compiled together with the rest of the sources.

Figure 55. Including the `Jai_Factory_Dynamic.h` inside user application.

```
// DynamicLoadSampleDlg.cpp : implementation file
//

#include "stdafx.h"
#include "DynamicLoadSample.h"
#include "DynamicLoadSampleDlg.h"

// In order to dynamically link to Jai_Factory.dll you will have to include the file below once in the project
// This declares the entry points for the DLL functions and creates stubs that works with Jai_Factory.h declarations.
#include "Jai_Factory_Dynamic.h"
```

But in order to make this work together with the standard `Jai_Factory.h` include file then it is necessary to define the symbol `JAI_SDK_DYNAMIC_LOAD` before the `Jai_Factory.h` is included.

Figure 56. Example of how to define `JAI_SDK_DYNAMIC_LOAD` symbol

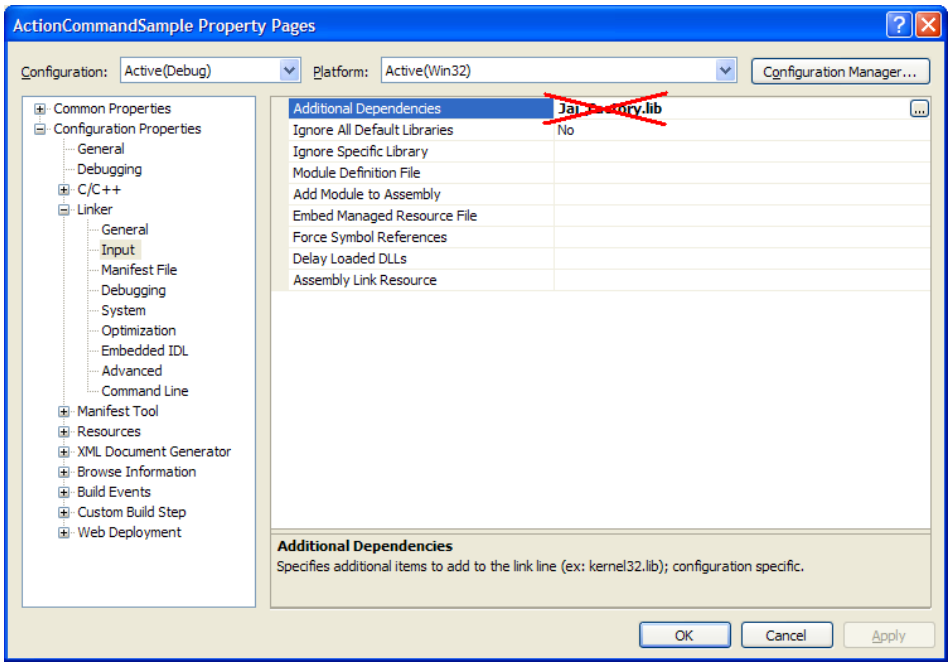
```
// DynamicLoadSampleDlg.h : header file
//

#pragma once

// It is necessary to define JAI_SDK_DYNAMIC_LOAD before including Jai_Factory.h if the Jai_Factory_Dynamic.h is going to
// be used for dunamically loading the Jai_Factory.dll!!!
#define JAI_SDK_DYNAMIC_LOAD 1
#include "Jai_Factory.h"
```

Finally it is necessary to avoid adding the `Jai_Factory.lib` to the list of linker input libraries.

Figure 57. Avoid linking with `Jai_Factory.lib`



When the user application is started then the `Jai_Factory.dll` will not be loaded until the function `J_Factory_Open()` is called. And the `Jai_Factory.dll` will be unloaded again by calling the `J_Factory_Close()` function.

8 Installing the JAI SDK in unattended/silent mode

If the JAI SDK needs to be installed from inside another installer then it is possible to add specific command line options to make the JAI SDK install as an unattended/silent installation.

The following example shows the command line options needed in order to do a quiet installation:

<JAI SDK installer>.exe /s /v" /qn [INSTALLFILTERDRIVERPROP=0] [REBOOT=Suppress]"

The options displayed within square brackets [] are optional parameters. The `INSTALLFILTERDRIVERPROP=0` will disable the installation of the Filter Driver (which will be installed by default) and the `REBOOT=Suppress` parameter will make the installer suppress the Reboot dialog.

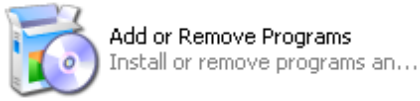
Note: It is very important to remember the quotes (") right after the **`/v`** *parameter and at the end of the command line! And it is also important to keep an empty "space" character right after the first quote and in front of the* **`/qn`** *parameter!*

So if for instance the installation of 64-bit JAI SDK version 2.1.0 needs to be unattended and the Filter Driver is not needed and the Reboot dialog needs to be suppressed, then the following command-line needs to be used:

JAI SDK x64 Version 2.1.0.exe /s /v" /qn INSTALLFILTERDRIVERPROP=0 REBOOT=Suppress"

9 Uninstalling the JAI SDK and Control Tool

The JAI SDK & Control Tool is uninstalled by using the Add or Remove Programs function found in the Windows Control Panel.



9.1 Uninstalling the JAI SDK in unattended/silent mode

If the JAI SDK needs to be uninstalled from inside another installer then it is possible to add specific command line options to make the JAI SDK uninstall as an unattended/silent uninstall.

The following example shows the command line options needed in order to do a quiet uninstall:

<JAI SDK installer>.exe /s /x /v" /qn [REBOOT=Suppress]"

Europe, Middle East & Africa
Phone +45 4457 8888
Fax +45 4491 3252

Asia Pacific
Phone +81 45 440 0154
Fax +81 45 440 0166

Americas
Phone (Toll-Free) 1 800 445-5444
Phone +1 408 383-0301