Release Notes for JAI Vision SDK

Last Updated: Sept. 7th, 2018

Latest Release Version: 3.0.7

General Notes:

1. The installation program should be run with elevated privileges (this is a MUST for Windows 8.x and Windows 10), or the installation may fail.
   a. Right-click on the installation executable in Windows Explorer (i.e. File Manager) and select “Run as Administrator”.
2. USB 3.0 cameras are not supported in Windows XP or earlier operating systems.
3. Windows 7 may require a reboot the first time a USB3 Vision device is attached to the JAI USB driver. Windows 8.1 and Windows 10 does not seem to require this.
4. USB discovery for non-JAI cameras in Windows 8.1 and Windows 10 might take10-20 seconds. It is almost immediate in Windows 7.
5. 32-bit applications are not guaranteed to run on 64-bit Windows.
6. Windows7 requires SP1 or later.

NOTE:

- **With regard to version 3.0.x, LAG performance is getting worse than version 2.1.6 because there is an issue with the Filter Driver of JAI SDK. Therefore, keep using 2.1.6 is recommended if use GEV cameras with LAG connection.**
- **When "Enable Display During Acquisition" in the settings dialog of the control tool is set to "False", Image recording function is also disabled.**

Modification Log:

Sept 7 2018: JAI SDK Version 3.0.7.0

1. Added new cameras.
2. Fixed bug that the phenomenon under certain condition that cannot convert image at C++ StreamThreadSample.

June 8 2018: JAI SDK Version 3.0.6.0

1. Fixed bug that the illegal image which occurs in case of specific condition when using GigE Camera.

Feb 28 2018: JAI SDK Version 3.0.4.1

1. Fixed bug that the Event Data is not displayed when using GigE Camera which have Event Control.
2. Modified phenomenon that a lot of logs appears in Visual Studio Debug Window when using chunk unsupported camera.

Jan 29 2018: JAI SDK Version 3.0.4.0

1. Fixed bug that the phenomenon that cannot acquire RGB12V1Paccked image at C++ Sample.
2. Fixed bug that the Frame Rate value becomes half value when using RGB12V1Packed image.

Nov 30 2017: JAI SDK Version 3.0.3.0

1. Fixed bug that "Cannot open stream Error" occurs when starting acquisition with GigE camera after application is forced termination.

Oct 6 2017: JAI SDK Version 3.0.2.3

1. Modified the RGB10p32 pixel format support.
   - 3-1. Added the support codes for the RGB10p32 pixel format, which was renamed from the RGB10V2Packed with GigE Vision 2.0, to the CCameraClass of the JAIFactory. It handles the pixel format name "RGB10p32" as an alias of the "RGB10V2Packed".
   - 3-2. Fixed the incorrect support codes for the RGB10p32 pixel format which were added to the CCameraClass of the JAIFactory.
   - 3-3. Fixed the incomplete support codes for remote pixel formats in the CCameraClass of the JAIFactory.
     These changes make the following JAIFactory API functions match their original specifications.
     J_Image_Get_Remote_PixelFormat
     J_Image_Get_Remote_PixelFormatName
     J_Image_Get_Compatible_PixelFormat
     J_Image_Get_PixelFormatFromName
2. Added modification for RGB10p, RGB12p, BGR10p and BGR12p PixelFormat (Additional modification for 3.0.2.1).
   - 2-1. Removed the incorrect Histogram and LUT support processes for the RGB10p, RGB12p, BGR10p and BGR12p pixel formats in the CHist::ClearHistogram and CLUT::LUT of the VisionLib, and modified the methods to make them return VL_INVALID_PARAMETER error for these pixel formats.
     This is because the pixel data organizations of the RGB10p, RGB12p, BGR10p and BGR12p are not cyclic in pixel units. The Histogram and LUT are should not be supported for these types of pixel formats.
   - 2-2. Added the Histogram and LUT support processes for the RGB10p, RGB12p, BGR10p and BGR12p pixel formats to the CHist::ClearHistogram and CLUT::LUT of the VisionLib.
     The Histogram and LUT should not be supported for these pixel formats originally, but these changes are added to satisfy customer demand.
3. Added modification for XPadding (Additional modification for 3.0.2.1).
   - 3-1. Fixed the incomplete XPadding support code for the BayerBG/GB/GR/RG12p pixel formats in the CImageConversion::FromRAWtoImageSimpleNoGain of the VisionLib.
   - 3-2. Changed the values for the XPadding to zeros with the image buffers allocated.
     This is because the XPadding is meaningful with the acquisition images and not with the conversion images.
   - 3-3. Added the XPadding support codes to the CImageConversion::Convert3Y, ExtractAlpha8bit, ExtractAlpha10bitPacked ExtractAlpha12bitPacked and ExtractAlpha16bit of the VisionLib.
     These codes are effective when the CImageConversion::ConvertImage is called with an acquisition images.
   - 3-4. Fixed the incorrect codes which calculate the pixel byte addresses from the pixel indexes in the below methods of the VisionLib.
     GetPFNC10BitRGBPackedValue
     GetPFNC10BitRGBPackedValues
     GetPFNC12BitRGBPackedValue
     GetPFNC12BitRGBPackedValues
     PutPFNC10BitRGBPackedValue
     PutPFNC10BitRGBPackedValues
     PutPFNC12BitRGBPackedValue
     PutPFNC12BitRGBPackedValues
   - 3-5. Added the XPadding support codes to the CHist::CreateHistogram() of the VisionLib.
     These codes are needed and effective when the CHist::CreateHistogram is called with an acquisition image.
   - 3-6. Fixed the incorrect XPadding support codes for all the pixel formats which were implemented into the CHist::CreateHistogram() of the VisionLib by the commit b7756e6.

The old codes had caused a System.AccessViolationException in Jai_FactoryDotNET.dll when the J_Image_CreateHistogram is called with an acquisition image.

3-7. Added the XPadding support codes to the CLUT::LUT() of the VisionLib.
These codes are needed and effective when the J_Image_LUT => CLUT::LUT is called with an acquisition image.

3-8. Fixed the incorrect XPadding support codes for all the pixel formats which were implemented into the CHist::CreateHistogram() of the VisionLib.
The old codes had caused a System.AccessViolationException in Jai_FactoryDotNET.dll when the J_Image_CreateHistogram is called with an acquisition image.

3-9. Added the XPadding support codes to the CLUT::LUT() of the VisionLib.
These codes are needed and effective when the J_Image_LUT => CLUT::LUT is called with an acquisition image.

3-10. Added the new member iImageStatus to the tIMAGE_INFO structure.
This is used to avoid conversion images in the XPadding support processes of the VisionLib.

3-11. Modified all the XPadding support codes of the VisionLib to avoid conversion images in their processes by referring to the tIMAGE_INFO.iImageStatus.

3-12. Added the new member iImageStatus to the J_tIMAGE_INFO structure.
This is used to avoid conversion images in the XPadding support processes of the JAIFactory.

3-13. Modified all the XPadding support codes of the JAIFactory to avoid conversion images in their processes by referring to the J_tIMAGE_INFO.iImageStatus.

3-14. Added the new member ImageStatus to the ImageInfo structure of the Jai_FactoryWrapper class and the support codes depending on it to the CCamera and CDevice classes.
These are needed just to avoid conversion images in the XPadding support processes of the VisionLib and JAIFactory and not useful in the JAISDK.NET.

4. Fixed the incorrect LUT processing code of the CHist::PutPFNC12BitMonoPackedValue in the VisionLib.
The old code had produced an image distortion in the BayerGR/RG/GB/BG12p images applying LUT processing.
5. Fixed the incorrect pixel byte address calculating and LUT processing codes of the CHist::PutPFNC10BitMonoPackedValue in the VisionLib.
6. Fixed the J_DataStream_GetStreamInfo.

6-1. The old codes had caused a System.AccessViolationException in Jai_FactoryDotNET.dll when the LUT processing is enabled to the BayerGR/RG/GB/BG10p images.
Fixed the mistake code in the J_Image_GetStreamInfo to call the GetTLWrapper()->m_PDSGetInfo instead of the J_DataStream_GetStreamInfo.
The old J_DataStream_GetStreamInfo call caused a unhandled exception in the J_Image_GetStreamInfo.

6-2. Modified the incomplete code of the GetTLWrapper()->m_PDSGetInfo call in the J_Image_GetStreamInfo.

7. Corrected typo in the StreamCallbackSampleDlg.cpp code in StreamCBFunc() under the chunk data part, line 486:

May 25 2017: JAI SDK Version 3.0.2.2

1. Modified INF files for installation of JAI USB driver to use AP-3200T-USB and AP-1600T-USB.
2. Modified the entire image conversion processes for the RGB10p, RGB12p, BGR10p, BGR12p pixel formats in the CImageConversion::FromRAWtoDIB and FromRAWtoImage of th VisionLib.

May 10 2017: JAI SDK Version 3.0.2.1

1. Added the XPadding support.
2. Fixed bug in the WndProc() of the VisionLib ImageViewWindow.cpp causing the problem that zoom width and height in case of WM_MOUSEWHEEL are doubled when a BayerFast interpolation algorithm is specified.

3. Added the zoom offset correction codes to the VisionLib, which solve the problem that the pixel locations of the zoom image are calculated with wrong x/y values. This case can occur when the image is zoomed out by the mouse wheel.
4. Added the support codes for the RGB10p, BGR10p, RGB12p, and BGR12p pixel formats to the JAIControlTool.

Dec 27 2016: JAI SDK Version 3.0.2.0

1. Fixed bug that GigE camera does not recognize with certain laptop PC.

Dec 20 2016: JAI SDK Version 3.0.1.4

1. Fixed bug remaining in the CImageConversion::FromRAWtoDIBEx() and FromRAWtoImageEx() of the VisionLib causing the problem that a BayerExtended conversion is performed in the Bayer images even if the BayerStandard interpolation algorithm is specified. (This is related issue with item 3 in version 3.0.1.0)

Dec 19 2016: JAI SDK Version 3.0.1.3

2. Internal Release.

Dec 16 2016: JAI SDK Version 3.0.1.2

1. Fixed additional bug for the item 3 in version 3.0.1.0 so that this modification also works to the recording images.

Dec 9 2016: JAI SDK Version 3.0.1.1

1. Modified INF files for installation of JAI USB driver to use GO-5100M/C-USB, and other future products.
2. Fixed bug that "Script Validation Error" occurs with items which has float node type during validate Script file.

Dec 5 2016: JAI SDK Version 3.0.1.0

1. Added the missing code returning J_ST_SUCCESS
   to JAIFctory J_Node_GetNumOfEnumEntries(). This fixes the problem of
   the J_Node_GetNumOfEnumEntries that always returns J_ST_INVALID_PARAMETER.
2. Fixed the bug in the VisionLib causing the problem that a BayerStandard conversion is always performed in the Bayer images regardless of the interpolation algorithm specified.
3. Fixed the bug in the JAIFactory causing the problem that the bHasChanged is always set to true when the call of the J_Factory_UpdateCameraList or CFactory::UpdateCameraList is the second or later one, in case two or more transport layers are available.

July 11 2016: JAI SDK Version 3.0.0.7

1. Fixed bug in Control Tool that does not display the device layer properties of the camera.

June 29 2016: JAI SDK Version 3.0.0.6

1. Fixed bug in Control Tool that prohibited MultiCast IP Address to be set. Now it is editable using standard point-notation.
2. The installer only deletes the …\JAI\SDK folder (and it's subfolders) during an uninstall if other folders in the …\JAI\ folder are existing.

June 13 2016: JAI SDK Version 3.0.0.5

1. Fixed bug in JAIFactory in Win32 where build failed to recognize the "JaiCLTL" as a valid transport layer.
2. Fixed bug the myLocalViewWindowCallback() in the source\JAIControlTool\JAIControlTool\JAIControlToolForm.cs was calculating a pixel XY position as 1/2 values in case of EColorInterpolationAlgorithm.BayerFast|BayerFastMultiprocessor.

April 22 2016: JAI SDK Version 3.0.0.4

1. Fixed bug in JAI Factory .NET where errors were generated when non-Gige Vision cameras were streaming and the Feature Tree Information tab was open.
2. Updated file version resource for the Filter Driver to 3.0.0.4.

April 20 2016: JAI SDK Version 3.0.0.3

1. Fixed bug in Filter Driver where BSOD occurred on some systems when ChunkData was enabled in GigE Vision cameras.

April 15 2016: JAI SDK Version 3.0.0.2

1. Minor updates to the SDK API documentation.

April 12 2016: JAI SDK Version 3.0.0.1 - Internal Release

1. Fixed bug in Factory .NET wrapper that prohibited the unlocking of JAI GigE Vision cameras in Factory mode.
2. Fixed bug in the GigE Vision Socket Driver where the ChunkLayoutID didn’t get updated in all acquisition buffers and hence prohibited update of the Chunk Data.
3. Fixed bug in Control Tool that prohibited MultiCast IP Address to be set. Now it is editable using standard point-notation.
4. Updated Control Tool user guide with description on how to use MultiCast on GigE Vision devices.
5. Fixed BSOD issue in GigE Vision Filter Driver because of Spin-lock deadlock inside the Filter Driver while sending Packet Resend telegrams.

March 25 2016: JAI SDK Version 3.0.0.0

2. The GigE Filter Driver now supports Windows 10.
    a. Created a new mode that allows other filter drivers (i.e. WireShark) to monitor GVSP traffic.
3. Implemented generic chunk support in the SDK.
    a. New SDK functions: J_DataStream_GetBufferChunkData, J_Camera_AttachChunkData, J_Camera_UpdateChunkData, J_Camera_DetachChunkData.
4. Fixed a bug in the installer that did not uninstall the filter driver properly (since version 2.1.5.5).
5. Fixed a bug in the filter driver that incorrectly counted corrupt images.
6. Corrected issues with the function J_Image_SaveFile, where data was not flushed to disk right away.
7. Modified the JAI USB3 Vision GenTL Producer to better handle acquisition when using twelve or more cameras that are triggered simultaneously.
8. Added text search capabilities to the GenICam feature and property grids in the JAI Control Tool.
9. Corrected the “TestMultiAcquisition” sample program to compile without errors.
10. Corrected text and bitmap images in 32/64-bit installations.

Feb 12 2016: JAI SDK Version 2.2.0.0 - Internal Release

6. Experimental chunk support for GigE devices.

Jan 19 2016: JAI SDK Version 2.1.7.1 - Internal Release

1. GigE Socket Driver - Fixed blank rows at top/bottom of image.
2. Implemented multi-camera support in the JAI Control Tool/SDK for RGBaXX images (to work with TVI multi-spectral camera).
3. Modified ImageConversion sample to work with RGBaXX images (i.e. support the TVI multi-spectral camera).

4. Vision Library:
    a. Corrected FromRAWtoImage function to set alpha channel to 255 for each pixel.
    b. Corrected IcMallocEx to initialize iBPP and handle PF_BGR8_PACKED. Now extracting alpha works with TVI camera.
5. Allow user to select GenTL producers during runtime in TestMultiAcquisitionGentl sample.

Dec 12 2015: JAI SDK Version 2.1.7.0 - Internal Release

1. Improved lag performance in GigE socket driver.
2. Modified GigE producer to return GenTL return codes instead of GigE return codes from EventGetData.

Nov 12 2016: JAI SDK Version 2.1.6.9

1. Improvement to GigE filter driver:
    a. Improve handling of buffers in the incomplete list by removing buffer from list head and adding it to the tail.
    b. Corrected handling of “dropped” buffers when “Send Corrupt Frames” is FALSE (previous version did not really drop corrupted buffers).
    c. Improve statistics count: Corrupt buffers are those with packet loss > 0.

2. Modified the values for the following pixel formats to match those in PFNC:
    a. J_PFNC_PIX_MONO10_PACKED
    b. J_PFNC_PIX_MONO12_PACKED
    c. J_PFNC_PIX_BAYGR10_PACKED
    d. J_PFNC_PIX_BAYRG10_PACKED
    e. J_PFNC_PIX_BAYGB10_PACKED
    f. J_PFNC_PIX_BAYBG10_PACKED
    g. J_PFNC_PIX_BAYGR12_PACKED
    h. J_PFNC_PIX_BAYRG12_PACKED
    i. J_PFNC_PIX_BAYGB12_PACKED
    j. J_PFNC_PIX_BAYBG12_PACKED

October 27 2015: JAI SDK Version 2.1.6.8

1. Modified GigE filter/socket drivers and the Vision Library to correctly handle RGB/YUV pixel formats.
2. Added support for non-streaming GigE devices that do not supply a timestamp.
3. J_Image_EnableParallelProcessing is now called during device enumeration instead of on a per-camera basis at tht start of streaming. The value is now saved to the settings file.

October 7 2015: JAI SDK Version 2.1.6.7

1. Jai_Factory_Dynamic.h: Fixed return code error in OpenJaiSdk.
2. .NET SDK: Modified UpdateDeviceList to allow use of AvalData CXP board (with AvalData SDK AZP-ACAPTF-01_V1300).

September 17 2015: JAI SDK Version 2.1.6.6 - Internal Release

1. Removed code to fill in SHA1 value for USB 3.0 cameras if the value is missing in the camera.
2. Corrected a bug that prevented 10-bit/12-bit packed pixel formats from being displayed in the control tool.

September 16 2015: JAI SDK Version 2.1.6.5 - Internal Release

1. Added support for gracefully handling case where a hang occurs if a USB 3.0 camera is disconnected while streaming.
2. Added support for image flipping RGB images in the Vision Library.
3. Corrected a bug sample project “ChildWindowSample” the used “WAIT_TIMEOUT” instead of “J_COND_WAIT_TIMEOUT” when examining the return value of “J_Event_WaitForCondition”.
4. Removed experimental support for enumerating AvalData CXP frame grabber.

September 13 2015: JAI SDK Version 2.1.6.4 - Internal Release

1. Experimental adjustments USB 3.0 producer to see if Pleora iPORT USB-CL device can be supported.
2. Experimental support for enumerating AvalData CXP frame grabber. Note that RGB modes are still not supported.
3. Signed the filter drivers using the new JAI Authenticode signing certificate for platforms up to Windows 8.1.

September 3 2015: JAI SDK Version 2.1.6.3

1. Fixed an issue with the filter driver which might cause a system crash if massive packet loss occured.
2. GigE filter/socket drivers: Added support for monitoring the #times a request is made for missing packets.
3. Corrected bug which required closing/opening the control tool after loading a camera settings script before the settings became active.
    a. Seen with CXP cameras and pixel formats.
    b. After loading the script the camera will be in "disconnected" state; the user must click the camera icon before operating the camera.
    c. Added more status bar text during script loading.
4. Replaced “generic” error codes with more meaningful return codes whenever possible, throughout the SDK.
5. Created new category in the .NET Camera class for GigE-specific items displayed in "Feature Tree Information" window of the control tool:
    a. Moved the following from the "Image Acquisition" category to the new "GigE Image Acquisition" category.
        i. "Missing packets in last image"
        ii. "Total missing packets"
        iii. "Enable packet-resend?"
        iv. "Pass corrupt frames from driver?"
    b. Added a new property "Number of resends requested".

August 21 2015: JAI SDK Version 2.1.6.2

1. Internal Release.

August 21 2015: JAI SDK Version 2.1.6.1

1. Fixed an issue with the filter/socket driver which might cause a system crash when switching from sequenced ROI mode to full image mode.
2. Added support to the Vision Library/SDK for separating parallel processing modes from Bayer interpolation algorithms.
    a. Previous versions used enumeration values (such as BayerStandard and BayerStandardMultiProcessor) to enable/disable parallel processing. This was confusing because the selection of the Bayer interpolation algorithm also affected the use parallel processing when displaying mono images. We have added a new property (the “Parallel Processing Mode” in the settings dialog of the

control tool, or the “J_Camera_EnableParallelProcessingMode” function in the SDK) to globally set the parallel processing mode to one of three values:

    i. Default – Original behavior. Set the bayer interpolation algorithms to one of the “…MultiProcessor” values to enable parallel processing. Affects both color and mono images.
    ii. Enabled – Enables parallel processing for all supported pixel formats, independent of Bayer interpolation algorithm setting.
    iii. Disabled – Disables parallel processing for all supported pixel formats, independent of Bayer interpolation algorithm setting.

   b. The enumeration for the Bayer interpolation algorithms (“”) in the SDK still contains the “…MultiProcessor” values for backward compatibility. Future versions of the SDK may remove them and just use the new “Parallel Processing Mode” alone to set the processing mode.

3. Added a new setting to the control tool “Enable Display During Acquisition”.
4. Modified installation programs to ask the user: "Would you like to delete all files in the Data folder?", referrring to the “…\JAI\SDK\Data” folder. This is where application-generated files are located (i.e. acquisition window size/position, etc.). If the user chooses “No”, the all files other than this folder will be deleted. Previous version left behind the filter driver files.
5. Fixed a bug in the control tool where the application would crash acquisition is started after changing to a new manifest entry on cameras with multiple manifests.
6. The installation applications now delete the “GigE Vision Filter Driver” entries from the Windows “Start Menu” .

July 16 2015: JAI SDK Version 2.1.6.0

1. Internal release.

July 13 2015: JAI SDK Version 2.1.5.5 **Released**

1. Added enumeration to disable image display in the contol tool during acquisition.

July 09 2015: JAI SDK Version 2.1.5.4

1. Eliminated memoy leak in SDK function J_Camera_LoadSettings.
2. Eliminated potential crash in J_Image_OpenStream by returning an error code if the internal call to vlOpenStream (from the Vision Library) fails rather than continuing execution.

July 07 2015: JAI SDK Version 2.1.5.3

1. Corrected warning messages in the control when using the track bar controls by rounding down track bar values instead of rounding up.

June 23 2015: JAI SDK Version 2.1.5.2

1. Corrected the VCMissingSamples sample project – version 2.1.5.1 mistakenly used an old version of the file “VCSampleMissingFramesDlg.cpp”.

June 22 2015: JAI SDK Version 2.1.5.1

2. Updated the "Users Guide” and “Getting Started Guide":
   a. Discuss how the user would choose between the GigE filter and socket drivers.
3. Clarify in the SDK documentation that in general both "J_DataStream_StartAcquisition" and "J_Camera_ExecuteCommand” must be called to start acquiring images.

June 21 2015: JAI SDK Version 2.1.5.0

1. Fixed possible out-of-order frames in GigE filter/socket drivers.
2. Improved performance in GigE filter/socket drivers when using two-channel LAG cameras.
3. Fixed memory leak in .NET CCamera class.

June 10 2015: JAI SDK Version 2.1.4.5 **Released**

1) Corrected version numbers of the 32-bit installer and the filter driver.
2) The build number (the fourth digit of the release number string) will now be included in the installer executable name and installer dialogs.

June 1 2015: JAI SDK Version 2.1.4.4

1) Removed memory leak if invalid width/height is read from camera in CImageViewWindow::PaintFrame.

May 31 2015: JAI SDK Version 2.1.4.3

1) Fixed Camera Link enumeration issue from changes made in version 2.1.4.1.

May 27 2015: JAI SDK Version 2.1.4.2

1) Fixed GigE enumeration issue from changes made in version 2.1.4.1.

May 26 2015: JAI SDK Version 2.1.4.1

1) Fixed resource leak in GigE and Camera Link producer
    a. Call DeleteCriticalSection in GCCloseLib.
2) Replaced deprecated std::strstream with std::stringstream.
3) Fixed remaining compilation errors in sample code regarding (char*/int8_t*) casting.
4) Control Tool: USB cameras now display "blinking" camera icon when enumerated but not selected by user.

May 19 2015: JAI SDK Version 2.1.4

5) Fixed installer to recognize if either versions 1.4x or 2.x SDK are already installed. Previous version handled only version 2.x of the SDK.

May 12 2015: JAI SDK Version 2.1.3.5

1) Corrected the declaration for Jai_FactoryWrapper.EFactoryError J_GenTL_TLGetInfo to use UIntPtr for the “Size” parameter instead of UInt32.
2) Fixed installer so that if an older version of the SDK is already installed, a warning dialog will appear, and then the installer will exit rather than overwrite a current installation.
3) Corrected C++ sample StreamThreadSample to query the block ID using a 64-bit sized variable instead of 32-bits.

April 23 2015: JAI SDK Version 2.1.3.4

1) Fixed saving AVI files for non-GigE cameras. Needs checking with all camera types.

April 22 2015: JAI SDK Version 2.1.3.3

1) Added instructions to uninstall previous SDK versions before installing new ones to the General Notes section of this file.

2) The camera link producer will automatically retry reading the XML file from the camera at 9600 baud if it fails when using a higher baud rate.
3) .NET Wrapper CCamera class uses high res timer (StopWatch class) to make FPS measurements more accurate.
4) Added a space character before "fps" for status bar display.

April 20 2015: JAI SDK Version 2.1.3.2

1) Fixed bugs with J_Datastream_GetBufferInfo\ J_Event_GetData that ignored the size parameter and or returned incorrect results.

April 8 2015: JAI SDK Version 2.1.3.1

2) Fixed buffer corruption issues for GigE cameras.
3) Fix in the .NET wrapper that prevented the socket driver from being loaded.
4) Fix in filter driver DLL and GigE producer the reported zero for the block ID.
5) Modified the default value for the “PassCorruptFrames” value in the GigE filter/socket drivers to “TRUE”, which was the default for version 1.4x of the SDK.
6) Added 250ms Delay for camera link WriteRegister command.

March 24 2015: JAI SDK Version 2.1.3 **Released**

1) Corrected/expanded previous entries for this file.

Feb 10 2015: JAI SDK Version 2.1.2.2

1) Fix for "black lines" in GigE images.
2) Fixed bugs in enumeration when J_Factory_UpdateCameraList is called multiple times.
3) Removed code from the .NET Wrapper Factory class (see Version 2.1.0.5) which ignores GenICam device nodes that do not have the "DeviceControl" category in the node map (this was to prevent enumeration of “virtual framegrabbers). However the "DeviceControl" category is not mandatory, and valid devices were not being enumerated if this node was not implemented.

Feb 10 2015: JAI SDK Version 2.1.2 **Released**

1) Corrected INF files for installation of JAI USB driver to use GO-2000 (instead of GO-20000).
2) The installer now only deletes the …\JAI\SDK folder (and it’s subfolders) during an uninstall – other folders in the …\JAI\ folder are not deleted.

Feb 6 2015: JAI SDK Version 2.1.1.4

1) Removed message box notification for FUnlock warnings in control tool - messages are now placed in log (as warnings, not errors).
2) Added ability to handle both 4 and 8 byte register read/write from the RegisterAccessForm in the control tool.
3) Camera Link: reverted back to original code from version 2.1.0.3:
    a. WriteRegister adds a 500ms delay after the register write, and performs a “write confirmation” if baud rate is changed.
    b. Three attempt are made at reading the camera’s XML file.

Feb 2 2015: JAI SDK Version 2.1.1.3

1) Fixed a bug in the Factory’s UpdateDeviceList: it no longer stops enumeration if an enumeration error occurs. This was found initially when enumerating interfaces in Win 8.1 laptops, where a “Microsoft Hosted Virtual WiFi Network Adapter” with the same MAC address as the actual adaptor caused the enumeration to stop, after which no USB or CXP devices were ever found.
2) Set the thread priority of Vision Libraries’s stream thread to back to normal (see Version 2.1.0.5).

3) Added missing _cdecl to: J_Camera_GetInterfaceHandle, J_Camera_GetInterfaceHandle, J_Camera_GetTLHandle in Jai_Factory.h.
4) Crerated a new node map in the factory for the DataStream module so that potential GenICam features can be accessed from the factory.
5) The USB producer now catches any low-level streaming errors and returns an error code.
6) The max number of scheduled buffers per stream, as well as other advanced settings for the USB producer are now GenICam features (to be set only by JAI).

Jan 14 2015: JAI SDK Version 2.1.1.2

1) Internal release – additional experiments with handling 5 or more USB cameras.

Jan 8 2015: JAI SDK Version 2.1.1.1

1) Modified USB producer to handle 5 or more USB cameras.
   a. Uses hard-coded value for the max number of scheduled buffers per stream.

Jan 5 2015: JAI SDK Version 2.1.1

1) Fixed issue where the READMEM telegrams for our GEV producer incorrectly handles requests for non-DWORD aligned amounts of bytes (i.e. when the last few bytes of the XML data is not on a 4-byte boundary).
2) Improved logic regarding use of SkipImageDisplayWhenBusy in the .NET Wrapper's Camera class - should allow for better image display when using cameras with high frame rates.
3) Replaced the use of "WaitForSingleObject" with a boolean flag in the Vision Library's stream stream when determining if the thread has been terminated.
4) Replaced the use of "WaitForSingleObject" with a boolean flag in the Factory's dispatcher threads for when determining if the thread has been terminated.
5) Updated the libusbx/libwdi source files in the License\sources directory.

Dec 19 2014: JAI SDK Version 2.1.0.6

1) Fixed bug preventing multiple cameras from multiple hardware interfaces from working with the Vision Library simultaneously.
2) Added new Wireshark plugins for 1.12.1 and 1.99.
3) Updated license files.

Nov 17 2014: JAI SDK Version 2.1.0.5

1) The JAI SDK and control tool has been successfully used with the BitFlow Cyton CXP framegrabber using version 5.891 of the BitFlow SDK.
   a. When used with Ximilon GenTL Consumer 1.4 (1.4.3).
   b. The Cyton does not supply the pixel format when using DSGetBufferInfo during acquisition. Therefore, the C++ samples provided by the JAI SDK must be modified to query the camera's pixel format, then uses that value for subsequent image display or processing.
2) Known issue: using "custom acquisition threads" is preferable to using the built-in acquisition thread in the Vision Library when multiple cameras are used with USB or CXP cameras. This allows for user control of thread priority, the number of buffers announced to the acquisition engine and the type of image processing used.
3) Known issue: the number of buffers announced to the acquisition engine should currently be limited to 5 for USB cameras.
4) Fixed errors when saving/loading images to/from disk using SDK functions J_Image_SaveFileRaw and J_Image_LoadFileRaw.
5) Added support for building samples in VS2013. Note that some minor modifications to the code may be necessary (specifically related to casting "C" strings to the required type).
6) Added C++ sample "TestMultiCameraAcquisition" to allow acquiring multiple cameras from all hardware interfaces that support streaming. Note if the computer resources are limited or many cameras are attached, the frame rate may need to be lowered to permit acceptable image

display. Note also that using “custom acquisition threads” is preferable to using the built-in acquisition thread in the Vision Library when resources are low.

7) The “Getting Started Guide” and the “Users Guide” where updated to state the GenCP-Enabled Camra Link cameras are controlable both from the JAI Control Tool and through JAI SDK functions. Note that streaming is NOT supported for Camera Link cameras through the JAI Control Tool nor the JAI SDK.
8) Boosted thread priority of Vision Libraries's stream thread to above normal.
9) Added code to check for invalid Bitflow "virtual hardware" and ignore it (for example, the Bitflow CXP framegrabber). Currently looks for the "DeviceControl" category in the node map, and ignores the device if this node does not exist.

Nov 17 2014: JAI SDK Version 2.1.0.4

1) Fixed errors when enumerating generic GenTL devices from C++ using J_Camera_Open.
2) Fixed bug in the JAI GenTL producer for GigE where:
    a. The timeout passed to EventGetData was incorrectly used.
    b. DSRevokeBuffer did not delete the revoked buffer from internal collections. Subsequent calls to DSAnnounceBuffer would the return GC_RESOURCE_IN_USE.
    c. Timeouts in calls to EventGetData do not now generate first-time exceptions in lower-level code.
3) Fixed pixel format errors with AvalData CXP framegrabbert when using 8-bit RGB mode.
4) Silent will delete any filter driver files from a previous install.
5) Silent uninstall attempts to remove the entire JAI directory if possible (files created after installation may remain).
    a. All filter drivers should be uninstalled and deleted.
6) Corrected warning dialog text when trying to install the 32-bit installer on a 64-bit computer.
7) Camera Link:
    a. Use original code from GenICam source tree for handling manifests and writing registers.
    b. Removed a 500ms delay from the WriteRegister function after performing the register write. Also removes the “write confirmation” if baud rate is changed.
    c. Only one attempt at reading camera's XML file.

Sept 24 2014: JAI SDK Version 2.1.0.3

1) Fixed errors handling pixel format data from AvalData frame grabbers.
2) Fixed bug which prevent a 3rd-party GenTL Producer from being loaded if the environment variables GENICAM_GENTL32_PATH/GENICAM_GENTL64_PATH began with a semi-colon.

Sept 24 2014: JAI SDK Version 2.1.0.2

1) Fixed crash when using IPConfig.exe

2) Fixed possible crash when using Jai_FactoryWrapper.WriteRegister function.

3) Fixed bugs reading/writing file header information in J_Image_LoadFileRaw and J_Image_SaveFileEx functions.

4) Fixed bugs in GE filter/socket drivers when handling "Packed" pixel formats.

5) Fixed bug where control tool fails to "Scan for Cameras" after user clicks on a USB camera AFTER its power/USB cable has been removed.

6) Now uses environment variables GENICAM_GENTL32_PATH/GENICAM_GENTL64_PATH for the location of 3rd-party GenTL producer CTI files instead of hard-coded paths.

7) Created the environment variable "JAI_SDK_LIB_64" even for 32-bit installations. Now 64-bit builds should succeed on 32-bit computers.

8) Prevents the title bar in the control tool from displaying multiple copies of the text ("Factory Settings Mode Enabled!") when factory mode was enabled and the user searches for new cameras.

9) Corrected the display of the Model Name (in the tree-view of the control tool) for camera link cameras and CXP devices when using the AvalData frame grabber.

10) Added latest installation splash-screen image to "Getting Started Guide".

11) Corrected typo in Registry_32.xml and Registry_64.xml files for Bitflow framegrabber: use the text “CXP”, not “CPX”.

12) Made the generic GenTL functions publicly available again.

13) Removed obsolete power-on procedure for USB devices in the "Users Guide".

14) Updated proceedure for using Camera Link cameras with the control tool in the "Users Guide” and “Getting Started Guide".

15) Updated many samples to set framegrabber width/height/pixel-formats (if present) to match the camera settings.

    a. The following C++ samples work with the Active Silicon Firebird (Driver version 01.03.01):
        i. ChildWindowSample, DynamicLoadSample, ImageProcessingThreadSample, LensDistortionSample, LUTSample, SimpleImageDisplaySample, StreamCallbackSample, StreamThreadSample, ZoomImageDisplaySample.
    b. The following C# samples work with the Active Silicon Firebird (Driver version 01.03.01):
        i. ChildWindowSample, ConsoleApplicationSample, FeatureBrowserSample, ImageDelegateSample, LensDistortionCorrectionSample, SimpleImageDisplaySample, ZoomImageDisplaySample.

Sept 3 2014: JAI SDK Version 2.1.0.1 **Released**

1) Fixed problem with JAI Control Tool that would require using the Windows Task Manager to close the application if a GigE camera was disconnected while acquisition was in progress.

2) Added support for accessing multiple manifests stored in the camera using the new SDK function J_Camera_OpenEx.

July 31 2014: JAI SDK Version 2.1.0 **Released**

1) GigE Vision:

    a) Fixed problem with JAI GigE Vision GenTL producer that may prevent simultaneous detection of acquisition, connection and GigE events.

        i) SDK users must now always use J_Event_WaitForCondition / J_Event_WaitForMultipleConditions rather than WaitForSingleObject / WaitForMultipleObjects when waiting for EVENT_NEW_BUFFER events.

        ii) SDK users should now always use J_Event_WaitForCondition / J_Event_WaitForMultipleConditions rather than WaitForSingleObject / WaitForMultipleObjects when waiting for EVENT_GEV_EVENT_CMD or EVENT_CONNECTIONevents.

2) USB 3.0:

   a) Fixed potential crash of the JAI USB3 Vision GenTL producer when creating many application level threads before opening the JAI Factory.

3) Camera Link:

   a) The JAI Camera link GenTL producer is now disabled by default. When the producer is enabled by the user:

      i) Automatic detection of camera link devices during startup and rescan will be enabled by default.

      ii) Downloading of the XML data from the camera will be disabled by default.

   b) The JAI Camera link GenTL producer will now detect cameras on virtual COM ports.

   c) Automatic probing for camera link devices does not occur when the JAI Camera link GenTL producer is disabled.

4) SDK:

   a) Updated Jai_Factory.h and Jai_Factory_Dynamic.h to include the latest functions and remove obsolete ones.

   b) Added new helper functions:

      i) J_Camera_GetTransportLayerName

      ii) J_Camera_GetLocalDeviceHandle

      iii) J_Camera_GetInterfaceHandle

      iv) J_Camera_GetTLHandle

      v) J_Image_Get_PixelFormat

5) JAI Control Tool:

   a) The JAI SDK version number is now displayed in the Control Tool's About box.

6) Samples:

   a) Added call to OnBnClickedStopButton() in OnDestroy() to stop acquisition cleanly if cancel button is pressed.

June 20 2014: JAI SDK Version 2.0.20 **Released**

1) Installation program now optionally installs all JAI USB device drivers.
   a) Driver for non-JAI USB devices must be installed as before, using the control tool.

   b) A stand-alone USB installation program executable will also be included that may be used by customers if needed (See "Drivers\USB3_Driver\ JAI_x86.exe and JAI_x64.exe").

   c) Devices must be uninstalled using the Windows Device Manager

2) Fixed camera link issues preventing the XML data for multiple CL models from being properly displayed.
3) Fixed endian setting in the "Read\Write Register" dialog for USB devices (it will be set to little endian).
4) Corrected bug that did not allow multiple attempts at opening/closing the factory in the SDK.
5) Fixed a bug where the "Pause/Snap" button in control tool did not work for USB cameras.
6) CAM_INFO_SERIALNUMBER now returns correct value for USB cameras.
7) Enabled setting of JAI_USB3_Vision by default in both 32/64-bit versions of the registry files.
8) The SDK now includes both the Registery_32.xml and Registry_64.xml files in the installation.

9) The SDK now includes “Xerces-C_MD_VC80_v2_7_1.dll” in the JAI\SDK\bin\Win32i86 folder when using the 64-bit installer.
10) Editing values (using the text box) in the control tool now works the same as with the slider control – dependent nodes will be updated.
11) USB transport layer now handles DEVICE_INFO_SERIAL_NUMBER.
12) C++ samples modified to compile using VS2010.
13) Added support for making the control tool a "TopMost" window on top of any image windows.
14) If during acquisition the image buffer returns an invalid pixel format from a 3rd-party transport layer, the value read from the camera during initialization is used.
15) Corrected errors in pixel format handling:
    a) Modified FromRAWtoDIBEx to handle additional pixel formats.
    b) Corrected PF_MONO12_P conversion in FromRAWtoImage.
    c) Corrected even column handling in AdjustBayerFlipHorizontal.

March 06 2014: JAI SDK Version 2.0.17 **Released**

1) GigE socket driver now supports link aggregation.
2) First support of the Pixel Format Naming Convention (PFNC).
3) The user may select between three installation types for the USB 3.0 Driver:
    a) Long installation – Does not require a system reboot (if the Control Tool is run with administrator privileges), and 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 unplug/plug-in 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.

March 03 2014: JAI SDK Version 2.0.16 (internal release only)

1) The GigE filter driver now supports link aggregation.
    d) NOTE: the GigE socket driver does not support link aggregation.

Feb 15 2014: JAI SDK Version 2.0.15 **Released**

1) Windows 8 and 8.1
    e) The installation program MUST be run as administrator (or the installation may fail).
2) Filter Driver:
    a) Fixed bugs in filter driver that caused BSOD when changing between 32bit and 64-bit block IDs.
    b) GigE 2.0 support for LAG not supported.
    c) GevFD.dll: fixed bug getting STREAM_INFO… parameters from filter driver.
3) Camera Link:
    a) Concatenated camera model name with the serial port interface name to make sure two CL devices of the same type will be recognized.
4) SDK:
    a) Added support for all JAI pixel formats from VL.h (missing BayRG10Packed, for example).
5) CXP:

    a) Synchronize width/height values between CXP framegrabbers and camera. This allows some framegrabbers (such as AvalData) to display images without requiring the user to manually set values.
    b) HOWEVER, pixel format values are NOT synchronized – these (and possibly other values) may require the user to manually set the same values on the camera and the framegrabber (i.e. the "Local Device" layer) in the control tool.

6) USB:
    a) 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.

    b) Although Windows 8 and above come with native USB 3.0 drivers, the may NOT be adequate for some USB 3.0 controllers.

    c) Some USB 3.0 controller/driver packages that work well on one platform may not work well on another (i.e. 64-bit versus 32-bit windows).

    d) Most connection problems are related the USB 3.0 controller/driver.

    e) Added description in PDF documentation.
    f) Improved text in driver installation dialogs.

Jan 21 2014: JAI SDK Version 2.0.14 (internal release only)

1) Factory:
    a) Fixed bug in reading XML files from the camera (as opposed to reading ZIP files from the camera, which works correctly).
        i) Should now handle cases where data is NOT aligned on a DWORD.
2) USB:
    a) Added workaround to handle cameras with invalid value for SHA1 GenCP register, which is used for creation of camera's XML zip file.
3) Filter Driver:
    a) Fixed translation of network packet ID values for GigE 2.0 cameras.
4) Control Tool:
    a) Moved "Logging Properties" setting under the "Support" category.
5) Samples:
    a) Added JAI pixel format handling to StreamThreadSample sample.

Jan 12 2014: JAI SDK Version 2.0.13 (internal release only)

7) Added back missing header files to SDK library.

Jan 12 2014: JAI SDK Version 2.0.12 (internal release only)

1) Fixed log setting bugs: now individual logs can be correctly enabled/disabled.
    a) Default is to log to any available DEBUG WINDOW, not to file on disk.
2) Added correct version number to filter driver.

Jan 08 2014: JAI SDK Version 2.0.11 (internal release only)

1) Experimental filter driver to debug 64-bit block IDs.
2) Modified code to allow for successful compilation of SimpleImageDisplaySample sample.

Jan 08 2014: JAI SDK Version 2.0.10 (internal release only)

1) Added support to the register read/write dialog to set the register endian-ness (factory mode only).

Dec 09 2013: JAI SDK Version 2.0.9 (internal release only)

1) USB installer integrated into control tool.

2) First support for image streaming for USB3 Vision cameras.
3) Known issues:
   a) BSOD noted when changing between 32-bit and 64-bit block IDs.

Nov 25 2013: JAI SDK Version 2.0.8 (internal release only)
1) Initial support for multiple manifest files in the filter driver.

Nov 08 2013: JAI SDK Version 2.0.7 (internal release only)
1) Initial support for an installer for the JAI driver for USB3 Vision cameras (using standalone JAI_Usb_Driver_Installer_Win32.exe and JAI_Usb_Driver_Installer_Win64.exe programs).

Oct 17 2013: JAI SDK Version 2.0.6 (internal release only)
1) Signed the filter drivers using the new JAI signing certificate.

Oct 11 2013: JAI SDK Version 2.0.5 (internal release only)
1) Initial support for U3V cameras (enumeration only).

Oct 04 2013: JAI SDK Version 2.0.4 (internal release only)
1) Added support for setting GevControlTimeout and GevControlRetryCount in the settings dialog.
2) Fixed the size anchors for controls on camera link dialog.
3) Added support for logging debugging output for the GenAPI, GenTL and CLProtocol modules (see “Logging Properties” in the settings dialog).

Sept 09 2013: JAI SDK Version 2.0.3 (internal release only)
1) Uses private build of Genicam 2.4 (to prevent conflicts with any other version of Genicam installed on the host computer).
2) First attempt at support for 64-bit block IDs in the filter driver. However, the driver is unsigned for the 64-bit platforms

Aug 12 2013: JAI SDK Version 2.0.2 **Released**
1) Known issues:
   a) If a framegrabber has been configured with “virtual” COM ports, both the custom name the and virtual serial port name may be detected when probing for camera link devices using the JAI Control Tool, when only one of the two is may be usable.
   b) Modified the SDK to handle the case where camera link devices are still detectable even if the uninstallation of third-party software incorrectly deletes the GENICAM_CLPROTOCOL Windows environment variable.
2) Added the following to the “GigE Transport Layer” settings in the Settings dialog:
   a) Preferred Driver Type
   b) Preferred Device Access Mode
   c) Enable Automatic Force IP
   d) Enable Subnet Conflict Warning?

Aug 5 2013: JAI SDK Version 2.0.0 Released
1) Documents updated to reflect the release of JAI SDK Beta 2.0.0.
2) The JAI Control Tool is now a generic GenTL 1.3 consumer that can load third-party GenTL 1.3 producers, allowing support of other transport layers such as CoaXPress, Camera Link, USB3 Vision etc.
3) The JAI Control Tool now supports using the Settings dialog to modify (i.e. add/remove/edit) the registered transport layers loaded by the factory at runtime (one list each for 32/64-bit transport layers).
4) New GenTL 1.3 Producer DLL’s added
   a) A GigE Vision GenTL 1.3 producer DLL (JaiGevTL.cti) supporting JAI GigE cameras including new GigE Vision 2.0 features.
   b) A “Control Only” Camera Link GenTL 1.3 producer DLL (JaiCLTL.cti) supporting latest JAI CL cameras with GenCP protocol.
5) The Factory now supports:

a) Enumeration of compliant GenTL 1.3 producers
b) Backward compatible with original JAI Device Enumeration
c) Original sample code still works.

6) Support for Pixel Format Naming Convention 1.0
7) Supports a tree-structure display of the following GenTL modules (with full GenICam read/write support):
   a) Remote Device
   b) Device Layer
   c) Interface Layer
   d) Transport Device
8) The JAI Control Tool now supports a reverse tree-structure display of the GenTL modules:
   a) Transport Layer
   b) Interface Layer
   c) Device Layer
   d) Remote Device
9) Sample applications updated so they now shows how to utilize the new GenTL 1.3 API features.
10) The JAI Control Tool/SDK has been successfully tested with the Active Silicon FBD-4XCXP6 in PCI Express x8 CoaXPress Framegrabber. Please see http://www.activesilicon.com for more information.
11) To perform continuous acquisition using the function J_DataStream_StartAcquisition, set the "iNumImages" parameter to either 0 or 0xffffffffffffffff (ULONG_MAX).
12) The following C++ sample programs have been modified to work with both GigE and generic GenTL devices:
   a) ChildWindowSample
   b) DynamicLoadSample
   c) GreenCompensationSample
   d) ImageConversionSample
   e) ImageProcessingThreadSample
   f) LensDistortionSample
   g) LUTSample
   h) NodeEventSample
   i) OpenCVSample1
   j) ReadPixelSample
   k) RedCompensationSample
   l) SaveAndLoadSettingsSample
   m) SimpleImageDisplaySample
   n) StreamCallbackSample
   o) StreamThreadSample
   p) TransformSample
   q) VCAVISample
   r) VCSampleLIntensity
   s) ZoomImageDisplaySample
13) The following C# sample programs have been modified to work with both GigE and generic GenTL devices:
   a) AsyncImageRecordingSample
   b) AVISample
   c) ChildWindowSample
   d) ImageDelegateSample
   e) LensDistortionCorrectionSample
   f) NodeEventSample
   g) RedCompensationSample
   h) SimpleImageDisplaySample
   i) UserDrawSample
   j) ZoomImageDisplaySample
14) Samples Notes:

a) If a generic GenTL device does not support streaming, the controls related to streaming in the GUI will be disabled.
b) Image processing may be disabled for monochrome cameras in some samples.

January 11, 2013: JAI SDK Version 1.4.2 Beta 1 **Released**

1) New proprietary pixel format and color conversion code added in order to support YUV420pNV12 in `VL.dll`.

September 28, 2012: JAI SDK Version 1.4.1 **Released**

1) Problem in `Jai_factory.dll` fixed. On 64-bit systems the function `J_DataStream_GetBufferInfo()` for the `BUFFER_INFO_USER_PTR` resulted in stack corruption.

September 19, 2012: JAI SDK Version 1.4.1 RC 1 **Released**

1) Documents updated to reflect the official release of JAI SDK 1.4.1.
2) Problem in `Jai_FactoryDotNET.DLL` fixed: In the Control Tool it was not possible to set the values of any features of type `IIntConverter`, `IConverter` and `IIntReg`.
3) Problem in `VL.dll` fixed. Linear color interpolation added to the YUV 4:1:1 to RGB conversions.
4) The C# sample application `UserDrawSample` has been updated so it now correctly references the `Jai_FactoryDotNET.dll` from inside the default installation folder for JAI SDK.
5) The C/C++ sample application `HDRSequenceSample` has been updated. It is now possible to switch between different pixel formats without application crash. New "Auto Exposure" functionality added so the sample now can help the user to determine initial values for Exposure Time and Gain for the two sequences by sweeping the values and analyze the images.

July 02, 2012: JAI SDK Version 1.4.1 Beta 4 **Released**

1) Problem in `VL.dll` fixed. The YUV 4:1:1 to RGB conversion was incorrect. This error was exposed through either `J_Image_FromRawToImage()`, `J_Image_FromRawToImageEx()`, `J_Image_FromRawToDIB()` and `J_Image_FromRawToDIBEx()`.

June 14, 2012: JAI SDK Version 1.4.1 Beta 3 **Released**

1) Problem in `VL.dll` fixed. The "Standard" and "Standard Multi-Core" Bayer Color interpolation algorithm for 16-bit Bayer formats was not working correctly when converting to the JAI SDK internal 48bpp format using `J_Image_FromRawToImage()` and `J_Image_FromRawToImageEx()`. This problem was exposed if the user want to save a 16-bit Bayer image to disk as a TIFF image after doing color interpolation.
2) Problem in `VL.dll` fixed. The "Standard Multi-Core" and "Extended Multi-Core" Bayer Color interpolation algorithm would result in Handle and Thread leaks whenever acquisition was started and stopped. The problem was caused by a problem with the Visual Studio OpenMP libraries.

March 27, 2012: JAI SDK Version 1.4.1 Beta 2 **Released**

1) Problem in `Jai_Factory.dll` fixed. The function `J_Camera_LoadSettings()` did not return error if the settings file was wrong or missing.
2) The Filter Driver and the Socket Driver was updated so it now handles GVSP image packets with an error code other than "Success" correctly. A device might send out a packet with error code "Packet Unavailable" if a Packet Resend is sent by the driver so late that the cameras doesn't have the packet available any more.
3) The Filter Driver and the Socket Driver was updated so duplicate frames is avoided in case of massive packet loss. Before the driver sometimes would duplicate a frame (two frames with same Block ID) if packet resend is enabled and the camera was sending missing packets _after_ the frame already had been sent to the user application.
4) Documentation for the `J_DataStream_SetParam()` and `J_DataStream_GetParam()` has been updated with sample code.

February 9, 2012: JAI SDK Version 1.4.1 Beta 1 **Released**

1) Problem in GEVTL.dll fixed. When the Transport Layer enables Pending Ack in a device then the endianess for the value written to the GVCP Configuration register was wrong.
2) VL.dll updated so it now enables user applications to draw text and graphics overlay onto the View Window image display. The User-Draw handling is implemented as two callbacks to the View Window so it seperated User-Draw on top of the "image" (using Image pixel coordinates) as well as on top of the View Window itself (using screen pixel coordinates). The User-Draw overlay on the image will be scaled if the image is being scaled. The User-Draw overlay on the View Window is not scaled even if the image is being stretched to fit the size of the View Window. The User-Draw callback functions are enabled via the existing J_Image_SetViewWindowOption() and the callbacks are assigned using J_Image_SetViewWindowCallback().
3) VL.dll is updated so the Image drawing can be done using "double-buffering". This will remove any flickering of the User-Draw text and graphics overlay on the View Window. It is enabled/disabled using J_Image_SetViewWindowOption()and it should only be enabled if the User-Draw overlay on the View Window is used since the performance will be slightly decreased.
4) Jai_FactoryDotNET.dll is updated so it now exposes the new User-Draw overlay features.
5) The C/C++ sample application ReadPixelSample is extended so it also support the new User-Draw overlay feature. It will display sample text overlay as well as graphics.
6) New C# sample UserDrawSample added. It illustrates how to use the new User-Draw overlay features from .NET framework.
7) VL.dll updated. The Lookup-table processing using J_Image_LUT() has been optimized for speed. Especially the processing time of color images has been reduced.
8) VL.dll updated. The HDR processing has been extended so RGB8_PACKED, RGB10_PACKED, RGB12_PACKED and RGB16_PACKED pixel formats are now supported. The HDR output pixel format for all these is RGB16_PACKED.
9) VL.dll updated. The J_Image_FromRawToDIB() and J_Image_FromRawToDIBEx() converted the RGB16_PACKED pixel formats incorrectly so the Blue and Red channels were swapped in the DIB (ARGB) output.

September 29, 2011: JAI SDK Version 1.4.0 **Released**

1) Documents updated to reflect the official release of JAI SDK 1.4.0.
2) Problem in VL.dll fixed. The intermediate destination used for unpacking of the 10-bit BayerPacked and 12-bit BayerPacked pixel formats was wrong. This problem only happened if the width or height of the destination image was different from the width and height of the source image.
3) New feature in Jai_FactoryDotNET.dll: It is now possible to disable color interpolation for the 10-bit BayerPacked and 12-bit BayerPacked pixel formats in the Control Tool.
4) JAI SDK Software License Agreement updated so it now references all the 3rd party libraries distributed with the JAI SDK installers. Copy of license files for all the 3rd party libraries added to the ./doc/Licenses folder.

September 20, 2011: JAI SDK Version 1.4.0 RC 4 **Released**

1) Problem fixed in the C/C++ LensDistortionSample and the C# LensDistortionCorrectionSample. In order to speed up the lens distortion correction the samples now detects correctly whether the current PixelFormat is able to be processed without further conversions. The following PixelFormats can be corrected directly: Mono8, Mono10, Mono12 and Mono16. All other PixelFormats needs to be converted/normalized using one of the J_Image_FromRawToImage(), J_Image_FromRawToImageEx(), J_Image_FromRawToDIB() or J_Image_FromRawToDIBEx() functions.
2) Japanese translations of the updated "JAI Control Tool Users Guide" PDF-file and the updated "JAI SDK Getting Started Guide" PDF-file have been added.

3) Problem in Vision Library fixed. The Simple Bayer Color interpolation algorithm inside `J_Image_FromRawToImageEx()` didn't work correctly for 8-bit BayerRG when White-Balance was performed.
4) The Feature Properties display is modified slightly. In some cases where the camera XML-file included nested selectors, some of the features got accidentally removed from the list
5) Output format for the Printer and rtf-file output is improved. The name of integer selectors is now displayed and nested Categories is indented for increased readability. The output now uses the Visibility selection from the Control Tool (Beginner, Expert and Guru).
6) Problem in Control Tool fixed. The Control Tool now uses the image processing settings from the "Processing" tab when it creates AVI-files and saves the images from an Asynchronous recording session. The currently selected LUT and transformation functions (Rotate and Flip) will be applied the same way as it is applied for the Live Video display.
7) GigE Vision Control Protocol WireShark dissector updated so it detects the IP port used for the stream protocol and automatically add this GVSP dissector to the port. Version numbers for the dissector files changed to version 1.4.0.

September 7, 2011: JAI SDK Version 1.4.0 RC 3 **Released**

1) Problem fixed in the Control Tool Settings dialog. The "Reset to default" context menu did not work for the values that were defined as "unsigned integers".
2) Problem fixed in `Jai_FactoryDotNET.dll`. If the user selected the "Fast" color-interpolation algorithm in the Control Tool Settings dialog and performed an Auto White-Balance then the R-Gain and B-Gain result would be different from the result from all the other color-interpolation algorithms. Now the Auto White-Balance will be consistent.
3) The format for Printer, and rtf-file output of floating-point values has been changed so it is identical to the way the Control Tool displays these values. The format for IP-addresses and MAC-addresses has been updated so it is identical to the way the Control Tool display these values.
4) Minor change to the "Select Camera Settings filename" dialog that is displayed when the user saves the current camera settings to disk. The dialog now correctly displays all files with the .rtf extension.
5) Problem in `J_Camera_SaveSettings()` fixed. The output format is now compatible with the "Script" format that is created by the Control Tool when the user saves all camera settings using the "Save all Camera Settings to Script File" menu. The .script files can be loaded using `J_Camera_LoadSettings()` and files created using `J_Camera_SaveSettings()` can be run as scripts from the Control Tool.
6) The "JAI Control Tool Users Guide" PDF-file has been updated with description of the new functionality added to the Control Tool since JAI SDK version 1.3.0.
7) The "JAI SDK Getting Started Guide" PDF-file has been updated with information about using Microsoft Visual Studio 2010 and how to embed the JAI SDK installers into customers own installers and doing "Silent installs"
8) The Control Tool now utilizes the GenICam 2.x floating-point specific node properties that defines the display notation as well as display precision. If these properties haven't been specified in the XML-files from the cameras then the Control Tool will instead use default values that can be defined in the Control Tool Settings dialog.
9) Problem fixed in the Control Tool floating-point Slider Controls. If the mouse cursor was moved over the floating-point Slider Control then the current value would be sent to the camera regardless of the user was dragging the slider knob or not. This has now been fixed so the values will only be sent to the camera if the user drags the slider know with the mouse cursor.
10) The file naming of the Settings and Feature Tree Info rtf-files has been slightly modified so they follow the naming for other default file names in the Control Tool. The name now contains the Model Name and MAC-address as part of the name. The user can still select any name so this is just the default suggestion from the Control Tool.
11) The `HDR_AD081GESample` C++ sample has been updated slightly. It was not able to work with cameras with the latest SFNC naming for the trigger features. Now it correctly detects whether the firmware is old or new.

12) The Jai_Factory.dll internal version number has been updated to “1.4.0 build 1” to fit with the current release version.

August 25, 2011: JAI SDK Version 1.4.0 RC 2 **Released**

1) Problem fixed in `Jai_FactoryDotNET.dll`. If the `Width.Max` or `Height.Max` values for a camera are very high (like 32768) then the current `Width` and `Height node` value will be used instead of these theoretical maximum values. This problem is seen in some 3rd party cameras.
2) The `HDR_AD081GESample` has been updated so it doesn’t use a `manifest.xml` file any more. The `manifest.xml` will make the sample fail on 64-bit versions of Windows. All solution configurations have also been updated so the default Character Set is set to Unicode.
3) All OpenCV based samples have been updated so they all link with OpenCV version 2.1.0 libraries. The OpenCV libraries are now referenced in the main cpp-file instead of the project properties so it is now much easier to modify the samples in case a different/newer version of OpenCV is used. OpenCV version 1.x is no longer usable since the Camera Calibration sample is using OpenCV 2.x features.
4) New sample `ImageProcessingThreadSample` added to the installers. This sample illustrates how to handle image processing that takes longer time than the time between two consecutive frames. The image processing is done in a separate thread and the image display is done using standard C/C++ functions.
5) In then Control Tool settings dialog the “Image Acquisition->Buffer Count”, “Asynchronous Image Recording->Recording Count” and “Asynchronous Image Recording->Recording Skip Count” are now validated so negative values and the value 0 is no longer accepted as input (except for Asynchronous Image Recording->Recording Skip Count” which can still be 0).
6) Creation date for the “Properties” and “Feature Tree Info” printout now added in the Control Tool.
7) Caption and creation date for the “Properties” and “Feature Tree Info” rtf-file save and email now added in the Control Tool.
8) The automatic extension of the Heartbeat Timeout time by a factor of 100 inside the Transport Layer can now be enabled and disabled via a new environment variable `JAI_SDK_DEBUG=TRUE/FALSE`. The default behavior is disabled. If the `JAI_SDK_DEBUG` environment variable is not set to `TRUE` then this feature will be disabled. When this debug feature is enabled then the Transport Layer will automatically increase the Heartbeat Timeout by a factor of 100 in the camera if it detects that the execution is slowed down (typically cause by a user-breakpoint during a debug session). By extending the Heartbeat Timeout the Transport Layer will achieve that the camera is less likely to timeout and thereby close any open stream channels.
9) Problem in `J_Camera_SaveSettings()` fixed. If any of the enumeration entries were `Read-Only` (or `Not Available`) then the `J_Camera_SaveSettings()` would fail with a “GenICam Error”.
10) New sample `SaveAndLoadSettingsSample` added. This sample illustrates the use of `J_Camera_SaveSettings()`, `J_Camera_LoadSettings()` as well as `J_Camera_GetSettingsValidationErrorInfo()`.

July 21, 2011: JAI SDK Version 1.4.0 RC 1 **Released**

1) Version changed from 1.3.1 to 1.4.0 because the number of new features is not considered a sub-minor change but instead a minor change.
2) Problem in `J_Image_FromRawToDIB()` fixed. If the raw input pixel format is `BGRA32` or `RGBA32` then it is now correctly converted to BGR24.
3) Support for the GigE Vision `PENDING_ACK` command implemented in the Transport Layer. If a device support this feature then the TL will now be able to detect lost connections much faster since the general communication timeout no longer has to be set to the “worst case” time for executing a camera command but instead the device will be able to signal any delays as needed.
4) The `Jai_Types.h` has been updated so Visual Studio 2010 will be able to compile the sample application without warnings and error messages.

June 15, 2011: JAI SDK Version 1.3.1 Beta 13 **Released**

1) Wireshark dissectors recompiled for the latest released version of Wireshark version 1.6.x as well as latest unstable release version 1.7.0. The installers now contain dissectors for three different Wireshark versions 1.4.x, 1.6.x and 1.7.x.
2) The JAI Camera Control Tool is now updated so it is possible to save the "Feature Properties" and "Feature Tree Information" property grid contents to disk as rtf-files as well as easily send an email to camera support with the rtf-file attached. And it is possible to send all the camera settings to an email recipient as a script file as well.
3) Default value for the last parameter `iNumImages` in the prototyping for function `J_DataStream_StartAcquisition()` in `Jai_Factory.h` changed from `0xFFFFFFFFFFFFFFFFLL` to `0` to enable backward compatibility with Visual Studio 6 compiler. This has no influence on the SDK function since this parameter is not used in the Transport Layer and drivers.

May 25, 2011: JAI SDK Version 1.3.1 Beta 12 **Released**

1) FilterDriver updated so it is now possible to use the 64-bit FilterDriver from 32-bit application using WOW64.
2) All JAI SDK Installers updated: GenICam redistributable files now included directly into the installers. This enables the Full 64-bit installer to install both the 32-bit and the 64-bit JAI SDK and GenICam runtime files so WOW64 can be used for executing 32-bit applications on 64-bit Windows OS. 32-bit applications running on 64-bit Windows using WOW64 will be able to use the 64-bit FilterDriver seamlessly.

May 4, 2011: JAI SDK Version 1.3.1 Beta 11 **Released**

1) 16-bit Bayer pixel formats handling added to the enable/disable color interpolation inside the Control Tool and the .NET Wrapper. It was not possible previously to disable the color interpolation for 16-bit Bayer images.
2) Lens distortion correction now supports the three first terms in the Taylor series expansion around r=0 so it is now usable with fish-eye lenses. It is also possible to select the interpolation used during lens distortion correction. The possible choices are Bi-linear and no interpolation (nearest pixel). The distortion correction algorithm has been extended so it now can reuse a pre-generated undistort mapping in order to speed up the processing. This is enabled via new call to `J_Image_Processing(…,LENS_DISTORTION_MAP_INIT, pParameters)` and the map can be released again by calling `J_Image_Processing(…,LENS_DISTORTION_MAP_FREE,pParameters)`. This is demonstrated in the `LensDistortionSample` C/C++ sample application and the new `LensDistortionCorrectionSample` C# sample application.
3) New C/C++ sample `OpenCVCameraCalibrationSample` has been added to the SDK. This sample shows how to use the OpenCV library to determine the lens distortion correction parameters by doing a camera calibration using a Chess Board pattern. This sample requires OpenCV 2.0 or newer.
4) Problem with "FAST" and "FAST_MULTI" color interpolation handling in function `J_Image_FromRawToImage()` and `J_Image_FromRawToImageEx()` fixed. If the white-balance values were used then the color interpolation would return error if the destination buffer image width and height was smaller than the source buffer width and height. This would be the case for the FAST algorithms because the output image will be half the width and height.

April 1, 2011: JAI SDK Version 1.3.1 Beta 10 **Released**

1) New Image Processing function `J_Image_ProcessingEx()` added to the SDK. This is almost identical to the existing `J_Image_Processing()` function but this new function needs to be used when the image processing function require a separate Input and Output image (like Lens Distortion Correction).
2) Radial and Tangential Lens Distortion Correction added to the JAI SDK. This is now supported by the new `J_Image_ProcessingEx()` function.
3) New `J_Image_GetPixelEx()` function added to the SDK. This function makes it possible to read a "pseudo" pixel value that is located in between the actual pixels. The location is a floating-

point location and the pixel values are generated using bi-linear interpolation between nearest neighbor pixels.

4) The `J_Image_MallocEx()` has been extended so it is now possible to allocate an image buffer that has an identical size and pixel format as the “template” input image.
5) New `LensDistortionSample` added. This sample illustrates how to perform Lens Distortion Correction on both Monochrome and Color images.
6) Latest released GenICam® redistributables (version 2.1.1) added to the installers.

March 21, 2011: JAI SDK Version 1.3.1 Beta 9 **Released**

1) Problem fixed in `GEVTL.dll, GEVFD.dll and GEVSD.dll`. Sometimes the call to `J_Camera_Close()` would hang indefinitely when the internal Message Channel thread inside the Transport Layer was being stopped! This occurred because of a blocking recvfrom() call inside the thread. This has now been fixed by closing the blocking socket if this condition is detected when the camera connection is closed.

March 10, 2011: JAI SDK Version 1.3.1 Beta 8 **Released**

1) Problem fixed in `GEVTL.dll`. If a Read-Only camera connection had been established using `J_Camera_OpenMc(,,DEVICE_ACCESS_READONLY,)`, then a handle leak would happen when the camera connection was being closed later on using `J_Camera_Close()`! This is now handled correctly so no handles are leaked any more.

February 25, 2011: JAI SDK Version 1.3.1 Beta 7 **Released**

1) Problem fixed in the `Jai_Factory.dll` and `GEVTL.dll`. If a camera connection attempt using `J_Camera_Open()` failed (for instance if the camera was already controlled by another application) then a memory leak as well as multiple handles and a single thread would not be closed correctly! This is now handled correctly so multiple connection attempts can be performed without any problems.

February 16, 2011: JAI SDK Version 1.3.1 Beta 6 **Released**

1) Problem fixed in the Control Tool. The RichTextBox control used for Feature Properties Printing was erroneously made visible.
2) Problem fixed in `Jai_Factory.dll`: The function J`_Event_WaitForMultipleConditions()` did not work correctly.
3) A new C++ sample application `MultipleConditionSample` has been added to the installers. It shows how to use a single thread for all event processing (New Image, GigE Vision Events and Camera Connection events) using the `J_Event_WaitForMultipleConditions()` function. This sample also shows how the Camera Connection events can be used for automatically re-initializing the Data Stream so image acquisition is re-established after a camera has been disconnected and re-connected again.
4) Problem in `GEVTL.dll` fixed: If the camera connection is re-established after the camera previously got disconnected (camera rebooted or simply lost connection) then the Message Channel was not set up inside the camera so all GigE Vision Events would stop being sent to the Transport Layer.
5) If the camera connection is re-established after the camera previously got disconnected (camera rebooted or simply lost connection) then any previously created Stream Channels will automatically be reinitialized inside the camera so image streaming can be restarted by the user application by sending an `AcquisitionStart` GenICam command to the camera.
6) The `CCamera` class inside the `Jai_FactoryDotNET.dll` will now automatically try to restart acquisition inside a camera if the camera connection has been automatically reestablished by the Transport Layer. Acquisition will only be restarted if the acquisition has been started in the first place by either `CCamera.StartAcquisition()` or `CCamera.StartImageAcquisition()`.

January 27, 2011: JAI SDK Version 1.3.1 Beta 5 **Released**

1) Problem fixed in Jai_FactoryDotNET.dll: If acquisition is started and the number of acquisition buffers inside CCamera is too large for the amount of free memory then the OutOfMemoryException condition that will happen during buffer allocation is now handled in a way that the acquisition will be able to continue with a limited number of buffers.
2) Problem fixed in Jai_FactoryDotNET.dll: CNode objects can now both be selector and selectee at the same time. Before a node was either a selector or a selectee but for instance LUT features (like LUTIndex) is both a selector (for the LUTValue) and a selectee (selected by the LUTSelector).
3) The Control Tool has been modified so that the Feature Properties and the Feature Tree Information Grid contents can be printed out. “Print”, “Print Preview” and “Page Setup” is now supported.

December 30, 2010: JAI SDK Version 1.3.1 Beta 4 **Released**

1) The Vision Library has been extended so it now includes the Green-Compensation Image Processing algorithms. The functionality is similar to the existing Red-Compensation.
2) The structure declaration _J_tCOLOR_COMPEN and _J_tCOLOR_COMPEN_PARAMS used by the Color-Compensation Image Processing algorithms used with J_Image_Processing() has been modified in order to facilitate the implementation of the .NET wrapper version of J_Image_Processing(). The changes make it possible to Marshal the data structure from .NET to the underlying unmanaged function. The C/C++ sample RedCompensationSample has been updated so it uses the new structure definitions.
3) The J_Image_Processing() function has been added to the managed Jai_FactoryDotNET.dll. Overloaded versions have been implemented so it is easy to perform Red-Compensation and Green-Compensation from managed code. A new C# sample RedCompensationSample has been added to the installers.
4) Documentation has been updated with new example code for the new J_Image_Processing() in both C/C++ and C#.

December 21, 2010: JAI SDK Version 1.3.1 Beta 3 **Released**

1) Problem in GEVTL.dll fixed. If a camera is physically disconnected and the J_Camera_Close() is called immediately hereafter then a protection fault might occur inside the Message Channel thread for this camera.
2) A new generic Image Processing function J_Image_Processing() has been added to the Factory. This function is used for implementing different image processing algorithms inside the JAI SDK and currently a “Red-compensation” algorithm has been implemented. Documentation has been updated with this new function and a new Visual C/C++ sample RedCompensationSample has been added to the SDK.
3) The FilterDriver has been updated. The statistical counter for number of frames awaiting delivery was not updated correctly so if the user called J_DataStream_GetStreamInfo(STREAM_INFO_CMD_NUMBER_OF_FRAMES_AWAIT_DELIVERY) then the value sometimes would be incorrect the first time this function was called.
4) The Jai_Factory_Dynamic.h updated to reflect the API changes in Beta 3.
5) The HDR_AD081GESample sample has been slightly updated so it now works correctly with the released version of the AD-081GE cameras.

October 27, 2010: JAI SDK Version 1.3.1 Beta 2 **Released**

1) Problem in GEVTL.dll fixed. When the GigE Vision READMEM command was used and a camera replied with an error code (for instance “Invalid Address” or “Access Denied”) then the TL would loop and retry sending indefinitely. This has now been fixed so a proper error code is returned instead.
2) Problem in VL.dll fixed. If acquisition was started with a ROI smaller than then maximum Width and maximum Height then the Video Window would not be updated correctly.

3) The `Jai_FactoryDotNET.dll` and the Control Tool have been updated so they now support GigE Vision devices where the optional `TimestampTickFrequency` feature has not been defined.
4) The `Jai_FactoryDotNET.dll` has been updated so the case where `Width.Max` and `Height.Max` for a camera is set to a value >= 16K is now handled correctly. The node `PayloadSize` is now used for determining the size of all acquisition buffers before acquisition start instead of calculating the buffer size based on image size and pixel format.
5) Problem in `Jai_FactoryDotNET.dll` fixed. `CCamera.CreateSettingsScript()` would create script files that couldn't be validated and executed correctly if the camera XML-file included `iEnumeration` nodes where individual enumeration values was marked as "Not Implemented" via a `<pIsImplemented>` link. The `CCamera.CreateSettingsScript()` now correctly checks for "`IsImplemented`" and "`IsAvailable`" during the creation of the Settings Script. The `CNode.IEnumValue` class now contains new helper properties: `IsReadable`, `IsWritable`, `IsImplemented` and `IsAvailable`.
6) The GigE Vision FilterDriver has been updated so that "Blue Screen of Death" is now avoided when the FilterDriver is installed/uninstalled. The BSOD could in some rare cases happen if the systems had multiple NIC's installed or if the PC returned from Hibernation/Sleep mode.

September 28, 2010: JAI SDK Version 1.3.1 Beta 1 **Released**

1) The `VL.dll` has been updated with three new Bayer color interpolation algorithms ("Simple", "Simple Multiprocessor", "Fast" and "Fast Multiprocessor"). These new algorithms are optimized for speed by sacrificing the image quality and are therefore primarily intended to be used for the display of images – not for image processing. The "Fast" algorithm is up to ten times faster than the standard 3x3 Bayer algorithm and the "Simple" algorithm is about 2-3 times faster.
2) The `JaiFactoryDotNET.dll` and the Control Tool have been updated so it now supports the new faster Bayer algorithms.
3) A problem in `VL.dll` has been fixed. The background of the View Window was not correctly erased when the View Window was resized to become bigger than the actual image it displayed when "Stretch Live Video" was disabled.
4) The Control Tool has been updated so that any camera related changes inside the Settings dialog for the Control Tool will now be updated in all the cameras that are detected by the Control Tool. The user will be prompted before any changes are made to the camera object settings.
5) The WireShark dissectors have all been rebuilt for the latest released WireShark binaries (version 1.4.x). The Win64_x64 version of JAI SDK now includes 64-bit versions of the WireShark dissectors.
6) A problem inside the FilterDriver has been fixed. If the PC went into sleep/hibernation mode during image acquisition then a "Blue Screen of Death" would be displayed when the PC was returning from the sleep mode.
7) The `Jai_Factory.dll` has been modified so that whenever the factory is opened then it will automatically disable the sleep/hibernation in order to avoid that the connection between cameras and the PC is lost. When the Factory is closed then the sleep/hibernation will be restored to its previous state.
8) Problem in `GEVTL.dll` fixed. When the GigE Vision `WRITEMEM` command was used and a camera replied with an error code (for instance "Invalid Address" or "Access Denied") then the TL would loop and retry sending indefinitely. This has now been fixed so a proper error code is returned instead.

September 8, 2010: JAI SDK Version 1.3.0 **Released**

1) The Japanese versions of the Control Tool Users Guide and the Getting Started Guide have been updated.
2) The English Control Tool Users Guide has been updated to revision G. The description about known limitation during AVI-file creation has been updated and the available VfW codec in 32-bit and 64-bit version of Windows OS has been listed.

September 3, 2010: JAI SDK Version 1.3.0 Release Candidate 8 **Released**

1) The Jai_FactoryDotNET.dll has been updated. The internal copy of the last “raw” image from the camera is no longer dynamically allocated for each new image. It has instead been changes so that an earlier memory allocation will be “reused” in case the raw image size is identical to the size of the earlier images. This is to avoid the need for the Garbage Collector to run before the managed memory is made available again. During AVI-file creation the CPU load will normally be so high that the Garbage Collection will be postponed and thereby increase the memory pressure. This would lead to “out of memory” errors during the AVI-file creation. This problem has now been minimized.
2) The Control Tool has been updated so that the possible error messages that can be displayed during AVI-file creation (for instance in case of “out of memory” exceptions) have got more information that will help the user to better understand what to do about the reported errors.

September 1, 2010: JAI SDK Version 1.3.0 Release Candidate 7 **Released**

1) The Control Tool has been changed. When AVI-files are created then the memory allocated for the Bitmaps that are added to the AVI-file will be freed right away after it has been added. This makes the amount of memory required less because we don’t need to wait for the “Garbage Collector” to run before the memory will be available for the next Bitmap image to be created. This problem would occur if the number of recorded images is so high that the PC is running out of memory and that the image size is large (>5MB).
2) The .NET Wrapper dll has been updated so it will handle “Out of Memory exceptions” better if the PC is running out of memory during image recording.
3) The SDK documentation updated so the release month is now September instead of August.

August 30, 2010: JAI SDK Version 1.3.0 Release Candidate 6 **Released**

1) Error in the Control Tool was fixed. The Inter-Packet Delay calculator dialog has been changes so that manual input of values for the “Bandwidth %” and “New Value” requires the user to press the “Enter/Return” key before the value is registered.

August 25, 2010: JAI SDK Version 1.3.0 Release Candidate 5 **Released**

1) Errors in Control Tool fixed. The “Cancel” button on the message box displayed if the “Cursor Information” or “Zoom Navigation” window is manually closed did not work correctly. The size of the “Max Contrast” button on the “Processing” tab is increased so it will work on Japanese version of Windows.
2) Errors in `VL.dll` fixed.
   The `J_Image_FromRawToImage()`, `J_Image_FromRawToImageEx()` and `J_Image_LUT()` did not work with the 10-bit and 12-bit Bayer Packed pixel formats.

August 23, 2010: JAI SDK Version 1.3.0 Release Candidate 4 **Released**

1) Error in `GEVTL.dll` fixed. For GigE Vision 1.1/1.2 cameras the Transport Layer wrongfully assumed that the optional “Stream Channels Capability Register” always would be present and then it was unable to start streaming if the register read failed. This issue only affected 3rd party cameras – not JAI brand cameras.
2) Error in `GEVTL.dll` fixed. If a camera has been disconnected and the camera at the same time is being closed in this “disconnected state” then the “heartbeat supervision thread” sometimes would generate an “access violation” error. The problem has been fixed so the `J_Camera_Close()` will now wait until the “heartbeat supervision thread” successfully terminates before closing the camera object. The GigE® Vision Event message thread is updated with a similar fix in order to avoid potential problems like for the heartbeat thread.
3) Problems in `VL.dll` fixed. The `J_Image_Flip(pImageInfo,`*`J_FLIP_VERTICAL`*`)` would create artifacts at the end of some of the pixel lines if the lines were not 16-byte aligned. The 10-bit and 12-bit Bayer Packed pixel formats were not correctly handled in the `J_Image_Flip()` function because the resulting pixel format was not adjusted correctly.

4) Error in VL.dll fixed where the image display could sometimes halt if the image was resized at the same time as the camera frame-rate was very high (>200fps). Flickering of Image View Window during resize has been removed by optimizing the background erase message handling.
5) The J_Image_SaveFileEx() is now capable of saving images in “raw” format (without any header information). This deprecates the J_Image_SaveFileRaw() function.
6) The Control Tool has been updated so that the images can now be saved in all supported file formats (Tiff, Jpeg, Bmp, Jai, Raw).
7) Minor changes to Control Tool Users Guide document based on QA feedback. The sections describing the saving of the new proprietary .JAI “raw” image format as well as the possible AVI-file codec limitations has been updated.
8) New Event/Condition function J_Event_WaitForMultipleConditions() added so it is now possible to wait for multiple conditions inside the same thread. This makes it now possible to create a single thread that handles events for “new image”, “GigE Vision events” and “camera connection” events simultaneously. The online documentation has been updated with small sample code that illustrates how to use this new function.
9) It is now possible to zoom directly in the Video Window display using the mouse cursor. If the left mouse button is pressed and the mouse is dragged then the image will be zoomed accordingly. The zoom can be reset by clicking the right mouse button again. The direct mouse zoom is only possible if the “Stretch Live Video” display mode has been selected. This will work in the Control Tool as well as in user applications using the Video Window display in stretched mode. The Control Tool settings now include settings that enable the user to select if the Mouse Zoom should be enabled or not.
10) New functionality added to VL.dll so it is now possible to receive Image View Window events inside user applications. These events include “View Window Move/Size”, “Mouse Move”, “Left Mouse Button Up/Down”, “Right Mouse Button Up/Down” as well as “Zoom-position and Zoom-ratio changes”.
11) VL.dll and the Jai_Factory.dll have been updated so that Image View Windows can be configured using a new function J_Image_SetViewWindowOption(). The current View Window options can also be read using J_Image_GetViewWindowOption(). The functionality of the existing J_Image_SetImageOffset() has been extended with two new functions J_Image_SetImageOffsetEx() and J_Image_GetImageOffsetEx(). This makes it now possible for user applications to detect and handle changes in the Zoom-ratio and Zoom–position of an Image View Window.
12) The Control Tool has been updated so that the new “Image View Window events” is handled and it is now possible for the user to open a new “Cursor Information” window that will display the mouse cursor location in an image as well as reading the current pixel value at this location by clicking the left mouse button.
13) The C/C++ “ReadPixelSample” has been updated so it now shows how to use the new Image View Window events as well as setting the new View Window options.
14) Problem with displaying Integer and Float nodes as Slider Controls has been fixed. If the “Representation” attribute for an Integer or Float node has not been specified inside the GenICam™ camera XML-file then the default display would be “Pure Number” (which meant that there were no slider displayed). This has now been fixed so that the default display will always be like a slider control unless something else has been directly specified inside the GenICam™ XML-file from the camera.
15) Problem with the display of “Slider Controls” in Windows Vista and Windows 7 has been fixed. The slider “knob” was not updated while it was dragged by the mouse.
16) The AVI-file creation codec is now selected using ICCompressorChoose() instead of AVISaveOptions(). When this new function is used then the selection dialog only displays a list of compatible encoders for the input image type supported by the Control Tool. The error handling has also been updated so any AVI-file creation errors are display in clear text instead of an error code. A file locking problem has also been fixed for AVI-files that are temporarily created when an error occurs. The locks on these files have been removed so the same filename can be used right away.

17) The C# `AVISample` source code has been updated so it uses same code to create AVI-files as the Control Tool.
18) The `Jai_Factory_Dynamic.h` updated to reflect the API changes in RC4.

June 24, 2010: JAI SDK Version 1.3.0 Release Candidate 3 **Released**

1) Vision library `VL.DLL` has been updated. The Transformation and LUT functions have been updated so they now are multi-threaded/multi-core aware. This increases performance considerably on CPU's with multiple cores.
2) New conversion format added to `J_Image_ConvertImage()` and `J_Image_MallocEx()` so 24-bit RGB/BGR and 30-bit RGB packed (V1 and V2) images can be converted into an output image format where the three color panes are stitched together into a single image. The stitching can be either horizontal or vertical.
3) In processing tab in Control Tool the points used for Linear and Knee-point LUT creation can be reset by pressing and holding SHIFT key when either the Linear or the Knee button is pressed. This makes it easy to restore the points in case they are outside the visible range.

June 18, 2010: JAI SDK Version 1.3.0 Release Candidate 2 **Released**

1) Error in .NET Wrapper inside `CCamera` class.The connection delegate handling needed to be changed so it handled Connection Events correctly after they now are queued up.
2) Both 32-bit and 64-bit libraries now included in the Full versions of the installers. New environment variable `JAI_SDK_LIB_64` introduced so it is now possible to compile 32-bit executables on 64-bit systems and compile 64-bit executables on 32-bit systems. All C++ samples have been updated to reflect this change.
3) The `ChildWindow` C# and C++ samples have been slightly modified so they now also illustrates how to correctly handle resize of the child image view window using `J_Image_ResizeChildWindow()`.
4) `J_Factory_GetInfo(FAC_INFO_VERSION)` now returns a correct version string (1.3.0 build 1).
5) Version resources updated for `Jai_Factory.dll`, `Jai_Registry.dll` and `GEVTL.dll` so they now reflects the SDK version 1.3.0.
6) Unnecessary "NO_DATA" exception thrown by the Transport Layer was removed. This exception was thrown if a network interface had no IP-address associated.
7) The priority for the GigE Vision Event handling threads inside the TransportLayer has been changed to high priority in order to assure fast delivery of the events on systems with high load.

June 11, 2010: JAI SDK Version 1.3.0 Release Candidate 1 **Released**

1) The "JAI Control Tool Users Guide.pdf" and "JAI SDK Getting Started Guide.pdf" is updated to reflect the changes between JAI SDK version 1.2.5 and JAI SDK version 1.3.0.
2) Errors in `Fai_Factory.dll` functions `J_Camera_SaveSettings()` and `J_Camera_LoadSettings()` fixed.
    1. The Unicode handling of the settings file names was wrong so the file names were incorrectly assumed always being Unicode.
    2. The XML-file format namespace changed so the files created using `J_Camera_SaveSettings()` is now possible to load directly from the Control Tools "Run Script" menu.
    3. The `J_Camera_LoadSettings()` function is now able to use the script files created by the Control Tool.
3) The algorithm that determines the "Max Contrast" linear LUT values based on the Histogram contents has been modified slightly so it now uses a lower limit of 1% of the highest histogram entry value to determine the lower and upper range to be used in the LUT. The "Max Contrast" function is located on the "Processing" tab of the Control Tool.
4) The Control Tool now marks the features with a Wizard attached with a "*" in the name so it is now easier to determine where a Wizard dialog is available. The PacketSize editor is now added as a Wizards as well as the custom editor.

5) The Transport Layer `GEVTL.DLL` has been modified so the Camera Connection Event handling is conforming to standard event handling in the JAI SDK. This means that the Camera Connection Events are queued up – not only a snap-shot. The `J_Event_GetInfo()` will now be able to return the number of events queued up for delivery.
6) The C/C++ sample called ConnectionEventSample has been up updated so it illustrates how the Connection Events can be used for restarting the DataStream after a camera has been disconnected and reconnected again later on.

May 26, 2010: JAI SDK Version 1.3.0 Beta 11 **Released**

1) Error handling in TransportLayer improved so connections to devices where the MessageChannel capabilities are not implemented correctly will still be able to be created.
2) Message queue handling changed so there now is an upper limit to the number of pending events. If the application is too slow to handle the incoming amount of events then the queue size will be limited and an out-of-memory error will be avoided.
3) `J_Event_GetData()` will now return an error code if there are no events pending in the queue.
4) Documentation inside `Jai_Factory.h` updated for the event handling samples. It now call `J_Camera_UnRegisterEvent()`when the event supervision thread terminates.
5) `GigEVisionEventSample` updated so it now call `J_Camera_UnRegisterEvent()` when the event supervision thread terminates.
6) Handling of return status from `J_Event_GetData()` changed so it now works the same way for both DataStream events and GigE Vision events. If `J_Event_GetData()` is called and there are no events available then the return status is `J_ST_NO_DATA`. If any of the parameters to `J_Event_GetData()` are invalid then the function will return `J_ST_INVALID_PARAMETER`. Otherwise the return status will be `J_ST_SUCCESS`. The online documentation has been updated accordingly.
7) The sample application `StreamThreadSample` has been modified so the error code returned from `J_Event_GetData()` is handled correctly.

April 14, 2010: JAI SDK Version 1.3.0 Beta 10 **Released**

1) AVI-file problems have been reported for both the Control Tool as well as for the two JAI SDK samples. The latest Windows Media Player is no longer able to play the files that were created with previous versions of JAI SDK. This problem has now been solved and the samples have been modified so it is not possible to select the codec that is used when creating the AVI-files. In the Control Tool Settings dialog it is now possible to select if the user should be able to choose the AVI-file codec to be used as well as select if the AVI-file creation should be optimized when the camera is streaming 8-bit monochrome images.

April 8, 2010: JAI SDK Version 1.3.0 Beta 9 **Released**

1) Additional entry-points added in Jai_Factory.dll and Jai_Factory.lib in order to make these files backward compatible with applications built with JAI SDK version 1.2.5. The following entry-points has been added:
    1. `J_Image_OpenViewWindow`
    2. `J_Image_OpenViewWindowEx`
    3. `J_Image_SetViewWindowTitle`
    4. `J_Image_SaveFile`
    5. `J_Image_SaveFileEx`
    6. `J_Image_SaveFileRaw`
    7. `J_Image_LoadFileRaw`
    8. `J_Camera_SaveSettings`
    9. `J_Camera_LoadSettings`
    10. `J_Image_SaveLUT`
    11. `J_Image_LoadLUT`

March 31, 2010: JAI SDK Version 1.3.0 Beta 8 **Released**

1) Acquisition problem fixed when cameras are in `AcquisitionMode=SingleFrame` mode. This only happened for the SocketDriver and it was caused by the Firewall Traversal algorithm inside the TransportLayer. The problem has now been fixed in the TransportLayer `GEVTL.dll`.
2) It is now possible to save images from the Control Tool in the new "Raw Image Format (*.jai)". This new format is a proprietary JAI file format that preserves the raw image information such as pixel format, timestamp, width, height, offset X, offset Y, missing packets etc. These images can then later be loaded into JAI SDK using the library function `J_Image_LoadFileRaw()`.
3) Problem with display of Integer selectors in "HEX Display" mode in the Control Tool fixed.

March 18, 2010: JAI SDK Version 1.3.0 Beta 7 **Released**

1) LUT functionality updated so it is now possible to handle images from both RGB and Bayer color cameras.
2) New functions added to help create Histogram data from images. The following new functions were added:
    1. `J_Image_MallocHistogram()`
    2. `J_Image_FreeHistogram()`
    3. `J_Image_ClearHistogram()`
    4. `J_Image_CreateHistogram()`
    5. `J_Image_GetHistogramValue()`
3) The Processing tab in the Control Tool now includes the possibility to use LUT features on RGB and Bayer color cameras. It is also possible to display the histogram data in the same graph as the LUT data.
4) Since the GenICam 2.0.1 redistributables has been added then the default Integer node "Representation" is set to "Pure Number" by GenAPI (which tells the Control Tool that no Slider Control should be displayed). If the Representation is not specified (like in older versions of our cameras) then no Slider Control will be displayed anymore for the Integer Features inside the Feature Property Grid. The display of Slider Control can now be forced On inside the Control Tool by changing the Control Tool Settings `ForceIntegerDisplayAsSliders=true`.
5) When the Visibility level was changed in the Control Tool then it was only updated correctly for the selected camera. This problem has been solved now and all cameras will have correct visibility level.
6) New CSAutoExposureSample added. This C# sample illustrates how an "Automatic Level Control" can be implemented on a PC so a C3 cameras Gain and Exposure Time is automatically adjusted based on "average" values inside a region of the captured images.
7) The ARP cache will automatically be deleted for the network interfaces that includes cameras that will automatically be given an IP address using the GigE Vision ForceIP command. This will insure that the ARP cache is re-created correctly.

March 2, 2010: JAI SDK Version 1.3.0 Beta 6 **Released**

1) Problem inside the "Processing" tab in the Control Tool fixed. An exception was generated if a camera was manually disconnected and the LUT processing was re-initialized.
2) The "Inter Packet Delay Calculation" form was changes so it is now possible to manually set the Inter-Packet Delay value and the form then automatically calculates the maximum number of frames per second possible with the selected delay.

February 19, 2010: JAI SDK Version 1.3.0 Beta 5 **Released**

1) Improved error handling in TL and Factory. Communication errors between TL and devices are now logged.
2) Device communication errors are logged in the Control Tool Error log. This helps detect and fix any GenICam XML-file problems that leads to communication problems between devices and the Factory.
3) New GenICam version 2.0.1 now included.
4) New Lookup-table functions added. The following functions has been added: `J_Image_MallocLUT()`, `J_Image_FreeLUT()`,`J_Image_ConvertLUT()`, `J_Image_LUT()`, `J_Image_CreateGammaLUT()`, `J_Image_ApplyGammaToLUT()`,`J_Image_Creat`

eLinearLUT(), J_Image_CreateKneeLUT(), J_Image_GetLUTValue(), J_Image_SetLUTValue(), J_Image_SaveLUT() and J_Image_LoadLUT(). New C++ sample LUTSample added.

5) New Transformation functions added: The following functions has been added: J_Image_Flip() and J_Image_Rotate().New C++ sample TransformSample added
6) The Control Tool now has a "Processing" tab where it is possible to use the new LUT and Transformation functions when it is displaying images from the cameras.
7) The JAI SDK now handles 8-bit (ANSI) character strings and Unicode character strings correctly. The following functions has been changed so the now have a separate ASCII and a Unicode version:
    1. J_Image_OpenViewWindow -> J_Image_OpenViewWindowA and J_Image_OpenViewWindowW
    2. J_Image_OpenViewWindowEx -> J_Image_OpenViewWindowExA and J_Image_OpenViewWindowExW
    3. J_Image_SetViewWindowTitle -> J_Image_SetViewWindowTitleA and J_Image_SetViewWindowTitleW
    4. J_Image_SaveFile -> J_Image_SaveFileA and J_Image_SaveFileW
    5. J_Image_SaveFileEx -> J_Image_SaveFileExA and J_Image_SaveFileExW
    6. J_Image_SaveFileRaw -> J_Image_SaveFileRawA and J_Image_SaveFileRawW
    7. J_Camera_SaveSettings -> J_Camera_SaveSettingsA and J_Camera_SaveSettingsW
    8. J_Camera_LoadSettings -> J_Camera_LoadSettingsA and J_Camera_LoadSettingsW
    9. J_Image_SaveLUT -> J_Image_SaveLUTA and J_Image_SaveLUTW
    10. J_Image_LoadLUT -> J_Image_LoadLUTA and J_Image_LoadLUTW
8) All sample applications has been updated to use Unicode and they can now all be compiled in both ASCII and Unicode mode.
9) Filter Driver updated so it now counts missing packets correctly if the ImageLeader is missing.
10) New sample VCMissingFrames added to the installers. This sample illustrates how to get information about missing packets and frames.
11) New support file Jai_Factory_Dynamic.h added. This file helps dynamically load the Jai_Factory.dll instead of statically linking to Jai_Factory.lib. New Sample DynamicLoadSample illustrates how to use the Jai_Factory_Dynamic.h for dynamically loading Jai_Factory.dll.
12) New file type added to the J_Image_SaveFileEx() so it now can save a proprietary JAI RAW image format that holds information about the image buffer (such as OffsetX, OffsetY, SizeX, SizeY, PixelFormat, Timestamp , Number of missing packets…). This new file type can then later on be read by J_Image_LoadFileRaw() function and re-create the Buffer Info contents.
13) New J_Image_LoadFileRaw() function added to the Jai_Factory.dll. It is now possible to load previously stored RAW image files from disk.
14) New J_Image_LoadFileRaw() wrapper function added to Jai_FactoryDotNET.dll.
15) New macro J_BitsPerPixel added to Jai_Factory.h. This macro is supposed to be used when calculating image buffer size requirements based on Width, Height and PixleFormat. The older J_BPP which return Bytes Per Pixel cannot be used if the PixelFormat is one of the "Packed" formats since it will round of the number of bytes per pixel to an integer value. All samples updates so they now use the J_BitsPerPixel macro instead of J_BPP.

January 19, 2010: JAI SDK Version 1.3.0 Beta 4 **Released**

1) Error handling inside Jai_FactoryDotNET.CCamera.StartImageAcquisition(), Jai_FactoryDotNET.CCamera.StopImageAcquisition(), Jai_FactoryDotNET.CCamera.StartAcquisition() and Jai_FactoryDotNET.CCamera.StopAcquisition().
2) New function Jai_FactoryDotNET.CCamera.StopImageAcquisitionEx() added. It is identical to Jai_FactoryDotNET.CCamera.StopImageAcquisition() but will return an error code.
3) New Wizards added to the Control Tool to help set-up trigger based on GenICam Standard Feature Naming as well as Proprietary JAI naming. It tells the user how to set the trigger system into Continuous mode as well as into Software Trigger mode.

December 17, 2009: JAI SDK Version 1.3.0 Beta 3 **Released**

1) Error in the Control Tool NodeInfo form fixed when it displays Command nodes.
2) Unhandled exception fixed when trying to save camera settings on cameras with WriteOnly nodes present in the node tree.
3) New GenICam 2.1 runtime now used instead of GenICam 1.1.2.
4) Enumerations combo-box handling changed to accommodate new GenICam 2.0 handling of EnumEntries.
5) CCamera class now includes more properties with information from the last image acquired (BlockID, SizeX, SizeY, OffsetX, OffsetY, ImageSize).

December 10, 2009: JAI SDK Version 1.3.0 Beta 2 **Released**

1) Action Command samples changed so they open two cameras.
2) Transport Layer changes so that “old” Action Command acknowledge telegrams are flushed before new Action Command is sent.
3) New C++ and C# ZoomImageDisplaySample added to the installer. These two samples illustrate how to utilize the Pan- and Zoom-functions in the API.

December 9, 2009: JAI SDK Version 1.3.0 Beta 1 **Released**

1) GigE Vision 1.1 functionality added: Action Command and Firewall Traversal support added to Factory, Transport Layer, Drivers and .NET Wrapper.
2) It is now possible to broadcast GigE Vision Action Command telegrams and handle acknowledges back from the devices. This is implemented in Jai_Factory.dll as J_Factory_SendActionCommand(). In Jai_FactoryDotNET.dll it is both implemented as a wrapper function Jai_FactoryWrapper.J_Factory_SendActionCommand() as well as 2 new methods in the CFactory Class: CFactory.SendActionCommand() with 1 overloaded version without acknowledge telegram handling.
3) Automatic Firewall Traversal has been added for both the Message Channel and the Stream Channel inside the Transport Layer (GEVTL.DLL). When a device is opened and it is detected that the device is GigE Vision 1.1 compliant and Firewall Traversal is supported (GevMCSP register is implemented) then the Transport Layer automatically sends out “dummy” telegrams every 10 minutes to open up the firewall on the port used by the Message Channel telegrams from the cameras. The same is the case for the Stream Channel! Whenever a the Stream Channel is opened then the Firewall Traversal will be started if the device is GigE Vision 1.1 compliant and Firewall Traversal is supported (GevSCSP register is implemented).
4) Jai_FactoryDotNET.dll updated with new GenICam node access function that facilitates read and write of node values. Jai_FactoryWrapper.J_Camera_GetValueInt64(), Jai_FactoryWrapper.J_Camera_GetValueDouble(),Jai_FactoryWrapper.J_Camera_GetValueString(), Jai_FactoryWrapper.J_Camera_SetValueInt64(), Jai_FactoryWrapper.J_Camera_SetValueDouble(), Jai_FactoryWrapper.J_Camera_SetValueString() and Jai_FactoryWrapper.J_Camera_ExecuteCommand() added.
5) New “Send Action Command” menu item in the context sensitive popup menu in the camera tree-view area. This will open new “Send Action Command” dialog that can be used for sending GigE Vision Action Commands.

6) New C++ and C# ActionCommandSample samples added to the installer. These two samples illustrate how to use the new Action Command functions included in the JAI SDK.
7) All documentation for Jai_Factory.dll and Jai_FactoryDotNET.dll is updated with a description of Action Commands and the new API functions.
8) Japanese versions of the "Getting Started Guide" and the "Control Tool Users Guide" added.

September 26, 2009: JAI SDK Version 1.2.5 **Released**

19) Documentation updated to reflect September 23rd release
20) HDRSequenceSample updated so it support C3 Bayer Color cameras including White-Balance control.
21) New AutoExposureSample added. This sample illustrates how a "Automatic Level Control" can be implemented on a PC so a C3 cameras Gain and Exposure Time is automatically adjusted based on either "average" or "peak" values inside a region of the captured images.
22) The Control Tool now measures the time it takes to check for subnet conflict and prompts the user to switch this feature off in case it takes more than 1 minute to complete.
23) The FilterDriver has been updated so it handles internal buffer-underrun correctly.

September 16, 2009: JAI SDK Version 1.2.5 Release Candidate 5

1) New fields "AwaitDelivery" and "BlockId" added to ImageInfo structure in the Jai_FactoryDotNET.dll. This fixes a bug that made it impossible to save images that were recorded using the Asynchronous Image Recording function in the Control Tool.
2) New fields "iAwaitDelivery" and "iBlockId" added to J_tIMAGE_INFO in Jai_Factory.h.
3) New enumeration value BUFFER_INFO_BLOCKID added to J_BUFFER_INFO_CMD in Jai_Factory.h so the GigE Vision Stream Channel Block-ID can be read using J_DataStream_GetBufferInfo().
4) New enumeration value BLOCK_ID added to EBufferInfoCmd in Jai_FactoryDotNET.dll so the GigE Vision Stream Channel Block-ID can be read using J_DataStream_GetBufferInfo().

September 15, 2009: JAI SDK Version 1.2.5 Release Candidate 4

1) FilterDriver updated. Frame buffer overflow error fixed that would result in "Blue Screen of Death".

September 14, 2009: JAI SDK Version 1.2.5 Release Candidate 3

1) FilterDriver updated. Possible frame buffer overflow in case of missing Image-leader telegrams.
2) Jai_Factory.dll updated with new GenICam node access function that facilitates read and write of node values.
   J_Camera_GetValueInt64(), J_Camera_GetValueDouble(), J_Camera_GetValueString(), J_Camera_SetValueInt64(), J_Camera_SetValueDouble(), J_Camera_SetValueString() and J_Camera_ExecuteCommand() added.
3) C/C++ Samples updated so they are using the new direct GenICam node access functions. This makes the samples simpler to understand.
4) New sample that shows how to utilize the High Dynamic Range function with a AD-081GE camera

September 9, 2009: JAI SDK Version 1.2.5 Release Candidate 2

1) GigEVisionEventSample updated to recompile correctly under x64 operating systems.
2) Read/Write Register form changed to be modeless and the address and dialog location is persisted to the registry. Bug fixed so the address field no longer require "return" key to be pressed in order to accept new address value.
3) SocketDriver updated so the packet resend will now work for images with only few packets (less than 5).

September 3, 2009: JAI SDK Version 1.2.5 Release Candidate 1

1) Documentation updated for J_STREAM_INFO_CMD enumeration used in J_DataStream_GetStreamInfo() and J_Image_GetStreamInfo().
2) Sample applications GigEVisionEventSample and HDRSequenceSample updated to recompile correctly under x64 operating systems.

September 2, 2009: JAI SDK Version 1.2.5 Beta 5 Release

1) Jai_Factory.h and Jai_FactoryDotNET.dll has been updated with correct GigE Vision Stream Protocol pixel format values for 10-bit Bayer Packed and 12-bit Bayer Packed. The 16-bit Bayer Packed formats have been renamed to identical names without the "_PACKED" suffix according to latest GigE Vision Standard.
2) Documentation has been updated with all the Pixel Format defines.
3) Control Tool: Refresh button on Feature Properties tab will now invalidate all values and refresh the property grid itself. This will re-read all settings directly from the camera and refresh the values inside the GenICam cache.
4) Script Validation error fixed in Jai_FactoryDotNET.dll. It now handles variable Min and Max boundaries correctly for Selected features.
5) J_Camera_LoadSettings() and J_Camera_SaveSettings() has been added to the Jai_Factory.dll so it is compatible with the Jai_FactoryDotNET.dll CCamera scripting methods. It is then possible to for instance save a script from the Control Tool and later restore the camera settings from a user application written in C/C++ using the Jai_Factory.dll. The file format created and read by J_Camera_SaveSettings() and J_Camera_LoadSettings() is the same XML format as used by the CCamera scripting files. The J_Camera_SaveSettings() can further more be used for validating whether a given settings file is compatible with a camera or not. If any validation errors or warnings are detected in J_Camera_LoadSettings() then the function will return either J_ST_VALIDATION_ERROR or J_ST_VALIDATION_WARNING which are new status codes. To get detailed information about validation errors/warnings a new function has been added: J_Factory_GetSettingsValidationErrorInfo().
6) New versions of both SocketDriver and FilterDriver added. The new drivers now handle the loss of ImageLeader and ImageTrailer packets in the GigE Vision Stream Protocol. The older drivers would discard the whole Image if either of these framing packets were lost.

August 21, 2009: JAI SDK Version 1.2.5 Beta 4 Release

1) Factory: Fixed bug in the GenICam processing of GigE Vision EVENT_CMD and EVENTDATA_CMD telegrams.
2) Documentation has been updated with a new revision of the Getting Started Guide
3) The calculation of "telegram overhead" in the Inter-Packet delay wizard and in the .NET Wrapper has been fixed so it follows the GigE Vision Stream Protocol

August 13, 2009: JAI SDK Version 1.2.5 Beta 3 Release

1) Transport Layer: Event queue error fixed. Locking mechanism inside the event queue GetData function was wrong. GEVTL.DLL version number changed.
2) Factory: Added exception handling around all internal GenICam calls in order to avoid unhandled exception.
3) Documentation has been updated with more introductory information on how to use this SDK
4) New C++ sample added that illustrates the use of OpenCV and C3 camera sequences

August 6, 2009: JAI SDK Version 1.2.5 Beta 2 Release

1) Control Tool: Startup changed to display main window and "Please Wait" information faster during startup.
2) Transport Layer: GigE Vision Event acknowledge telegram error fixed. The "Length" field was set to 8 instead of 0
3) Transport Layer: Number of retries and timeout for GVCP commands changed for faster response and to avoid problems during cameras communication supervision.
4) Transport Layer: The Read and Write functions will terminate immediately if the connection to a camera is already lost. This will make the Control Tool recover faster in error situations where the camera connections are lost during image acquisition.

July 10, 2009: JAI SDK Version 1.2.5 Beta 1 Release

1) New "Event Condition" functions added as J_Event_CreateCondition(), J_Event_CloseCondition()… etc. These functions replace the

usage of Windows Events for the event handling in the SDK in order to make the code more portable between Windows and Linux.

2) New “High Dynamic Range” functions added. These functions make it possible to “fuse” two images with different exposure into a new image with higher dynamic range.
3) J_Image_ResizeChildWindow() function added.
4) Support for the new pixel-formats from the GigE Vision standard version 1.1 has been added.
5) Control Tool: Error fixed where the “Pause/Snap” button was not working properly.
6) Control Tool: Write-only features will keep the values set instead of 0 when they are being displayed in the Feature Property grid.
7) Control Tool: Execution of commands is now added correctly during script recording.
8) Control Tool: The Acquisition control buttons “enable state” now also reflects changes based on camera connection state.
9) .NET Wrapper: Camera class now caches values for “Write-only” nodes.
10) .NET Wrapper: New “Event Condition” functions added.
11) .NET Wrapper: New “High Dynamic Range” functions added.
12) .NET Wrapper: Commands can now be executed from within scripts.
13) .NET Wrapper: All internal threading is now utilizing the new “Conditions”
14) .NET Wrapper: Internal image buffer allocation is now 16-byte aligned in order to enable the use of the new HDR-processing functions.
15) All documentation has been updated with the new functions as well as better descriptions for all modules. More source code examples have been added as well.
16) Boolean nodes can now be set using J_Node_SetValueInt64() and read using J_Node_GetValueInt64()
17) Sample application has been updated with the usage of Condition
18) New samples have been added for handling GigE Vision events and on how to use the High Dynamic Range functions. New C# sample that shows how to use Async Image Recoding for multiple cameras has been added.

June 1, 2009: JAI SDK Version 1.2.4 Release

1) Documentation updated to reflect June 1st release.

May 28, 2009: JAI SDK Version 1.2.4 RC 1 release

1) .NET Wrapper: Explicit calling convention added for all DLLImport entries to correctly specify the __cdecl calling convention used by the Jai_Factory.dll.
2) Control Tool: It is now possible to specify multicast IP address for image data stream in the Control Tool settings menu so the cameras can be setup for multicast. And if a camera is opened as read-only it is possible to start multicast streaming as “slave”

May 7, 2009: JAI SDK Version 1.2.4 Beta 15 release

1) Control Tool: Additional tooltips added to the context sensitive menus.
2) Control Tool: Script Recording changed so current Selectors values always will be added if “selectees” are added without specifying the selectors before it.
3) .NET Wrapper: New properties added to the CNode class so all nodes will have a list of node-names of any selectors referencing them.
4) The Control Tool “settings” has been extended so ToolTip display can be disabled/enabled and so image acquisition timestamps can be displayed in milliseconds or ticks
5) Error fixed in Asynch Image Recording Replay Window display. On some systems the window did not display at all.

May 1, 2009: JAI SDK Version 1.2.4 Beta 14 release

1) Control Tool: New tooltips added to the main window buttons with additional information available.
2) When Asynchronous Image Recording is started then the Control Tool will try to calculate the maximum number of Images that can be recorded into available memory. The user will be

prompted if the number of requested images to be recorded is greater than the number that fits within available memory.

3) “Start Image Recording” menu will now be enabled correctly after save of recorded images as AVI-file.
4) “Start Image Recording” menu is now displayed correctly in Windows Vista.
5) The display of “IntReg” features has been corrected.

April 17, 2009: JAI SDK Version 1.2.4 Beta 13 release

1) GigE Vision EVENT_CMD and EVENTDATA_CMD handling implemented in Transport Layer (GEVTL.dll) and the Factory (Jai_Factory.dll). It is now possible to subscribe to these event via the J_Camera_RegisterEvent(hCamera, EVENT_GEV_EVENT_CMD,…) function. The EVENT_GEV_EVENTDATA_CMD event type has been deprecated and both event types are now handled via the EVENT_GEV_EVENT_CMD event type. The structure J_EVENT_DATA_GEV_EVENT_CMD used to get the event data has been updated accordingly including new m_EventDataLength and m_EventData byte array members.
2) Documentation for J_Event_GetData() and J_Event_GetInfo() has been updated including new sample code.
3) The Jai_FactoryDotNET.dll has been updated with GigE Vision EVENT_CMD and EVENTDATA_CMD handling according to the changes in Jai_Factory.dll. A new delegate (CCamera.GEVEventDelegate) has been added in the CCamera class. This will be called with the corresponding data from the GigE Vision Event (both EVENT_CMD and EVENTDATA_CMD) every time a new event is received from the Jai_Factory.dll.
4) A new Node event delegate has been added to Jai_FactoryDotNET.dll. This delegate will be called for every time a node callback has been called from GenICam node tree. All nodes with an EventID associated will automatically get a callback registered by the CCamera class. This makes it possible to update a GUI based on GigE Vision events handled by the GenICam node tree.
5) GenICam version 1.1.2 files is now included in the distribution.
6) The Camera Control Tool has been updated so it now subscribes to the GigE Vision events and add a new entry into the log (both the log window and in the log-file) whenever a GigE Vision event is received. The standard GigE Vision event types are interpreted and displayed directly in clear text (such as TRIGGER, EXPOSURE_START,…).
7) The Camera Control Tool GUI has been modified with new icons on the tab-pages.
8) All Integer features can be forced to be displayed as Hexadecimal notation (for instance: 0x0123abc) by setting “Settings->Feature Properties->HEX Display = True” in the settings dialog. If the Integer has got a negative minimum value then this feature will be displayed as decimal instead.
9) GenICam “representation” attributes now supported in the Control Tool. If an integer value has got the representation set to “HexNumber” then the integer will be displayed using Hexadecimal notation. The Slider Control used for changing Integer values will also display min/max/value using Hexadecimal notation.

March 31, 2009: JAI SDK Version 1.2.4 Beta 12 release

1) Access mode will be updated every time the CNode.EnumValues are read. This is necessary if pIsAvailable/pIsImplemented/pIsLocked is used for any of Enumeration entries and the access mode will change based on the values of other nodes. This change only affects Jai_FactoryDotNET.dll.
2) Asynchronous Image Recording to memory added in Jai_FactoryDotNET.dll and in the JAI Control Tool. There have been added methods and properties to control the recording in the managed dll and the Control Tool is also able to save the recorded images as RAW, Images (tiff, jpeg and bmp) as well as an uncompressed AVI file. It is also possible to “Replay” the recorded video in a separate Video Window.
3) New C# AsyncImageCaptureSample added to demonstrate the usage of the new Asynchronous Image Capture feature.
4) Getting Started Guide added to the documentation.
5) Icons added on the feature tab-pages.

February 24, 2009: JAI SDK Version 1.2.4 Beta 11 release

1) GetFactoryErrorString() method added to the Jai_FactoryDotNET.dll. It will convert a EFactoryError enumeration value returned from the J_xxx() methods into a description string.
2) MessageBox messages and system log messages added for the “Factory Settings” context menu items.
3) Status handling of “Factory Settings” context menu items changed so the status reflects the current “Factory Settings” mode correctly
4) Save In All Areas now also saves in Factory Area as well as in all User Areas
5) Unlock is automatically sent if the user switch to “Factory Setup” mode and whenever a camera is opened while the Control Tool is already in “Factory Setup” mode.

February 23, 2009: JAI SDK Version 1.2.4 Beta 10 release

1) This release note has been added to the JAI SDK installers.
2) Documentation updated so the code samples are not truncated in the PDF-files. This is done by reformatting the sample code.
3) GenICam version 1.1.1 RC2 included in the installation
4) New C# AVISample has been added to the Installers. It demonstrates how to create an AVI file from the image stream.
5) Problem with 10-bit Bayer color-interpolation algorithm for multi-core fixed.
6) J_Node_GetProperty() renamed to J_Node_GetPropertyByName() in Jai_FactoryDotNET.dll so it works correctly with Jai_Factory.dll
7) CNode.Presentation added to represent GenICam Presentation property value (such as “HexNumber”, “Linear” and “Logarithmic”) in Jai_FactoryDotNET.dll
8) Display of Integer node values in HEX notation has been added to Control Cool. The values will be displayed as HEX if the CNode.Presentation is set to HexNumber or if the “HexDisplay” is set to “true” in the Control Tool settings.
9) New “Factory Settings” context menus added to the Camera Tree View (Unlock, Save in Factory Area, Save in All Areas and Read/Write registers)
10) New “Direct Register Access” form added in Factory Settings mode only.

February 9, 2009: JAI SDK Version 1.2.4 Beta 9 release

1) “Refresh” button added to the Control Tool so all settings can be re-read by the user without moving the Property Grid slider.
2) Prefix for “top level” categories added like “a) <first category>”, “b) <second category>”… in order to avoid the order to be changes by the Property Grid in the Control Tool. Now the order will be determined by the order inside the Camera XML-files

February 3, 2009: JAI SDK Version 1.2.4 Beta 8 release

1) Error handling in J_Camera_OpenMC() has been updated so it doesn’t return J_ST_SUCCESS if it is called with an illegal Camera ID string.
2) New Slider controls in Control Tool for integer and float values that is able to handle 64-bit integers. Required by TM/TMC-series GigE Vision cameras
3) New export of camera settings to “script file” in the .NET wrapper
4) Export of settings to script file implemented in Control Tool via the “script” menu. It is now possible to store all camera settings to disk and later restore them using the “Run script” functionality of the Control Tool (and via the .NET wrapper CCamera class as well!)
5) Problem with minimized video windows in Control Tool fixed
6) Changes to the “script validation” functionality in .NET wrapper. It now treats the “out of range” values as warnings – not as errors that prohibit the script to be run

January 27, 2009: JAI SDK Version 1.2.4 Beta 7 release

1) Installers has been updated so the Settings can now be saved correctly in the Control Tool

January 26, 2009: JAI SDK Version 1.2.4 Beta 6 release

1) Installers has been updated in order to work with hidden Windows Vista folder structures

2) All the manifests used for “Requesting Administrator level execution” under Vista in all samples have been removed as well as in the Control Tool and the Persistent IP Configuration Tool.

January 25, 2009: JAI SDK Version 1.2.4 Beta 5 release

1) It is now possible to run the JAI SDK with Filter Driver without running the application from an “Administrator” account. This is made possible by changes in the Filter Driver
2) Installers has been updated to include changing security settings for the installation folders in order to write intermediate files at runtime

January 23, 2009: JAI SDK Version 1.2.4 Beta 4 release

1) Speed of the Filter Driver has been improved with around 12% over the older version.

January 20, 2009: JAI SDK Version 1.2.4 Beta 3 release

1) New support for IRegister, IIntReg and IFloatReg nodes has been added via the following functions: J_Node_SetValueRegister(), J_Node_GetValueRegister(), J_Node_GetRegisterLength () and J_Node_GetRegisterAddress()
2) New Register access wrapper functions added to the .NET wrapper dll and the CNode class has been updated as well.
3) In the Control Tool it is now possible to Load and Save the LUT values directly from/to files via these new features

January 19, 2009: JAI SDK Version 1.2.4 Beta 2 release

1) Fix of bug in the display of “nested selectors” where some features would show up multiple times in the feature tree.

January 15, 2009: JAI SDK Version 1.2.4 Beta 1 release

1) Exceptions that occurred in TL and Factory has been resolved
2) Vision Library and Socket Driver had to be updated in order to handle 10-bit and 12-bit Bayer Packed format, which had been announced in the latest GigE Vision.
3) Problem fixed in control tool, where "Start acquisition" and some other buttons were grayed-out if white-balance accidently got values above 4.0.
4) New function J_Image_SaveFileRaw added in the Jai_Factory.dll. It makes it possible to save the raw image contents received from the cameras as a binary file – without any “image file” header.
5) All Wireshark dissectors has been updated based on the GigE Vision 1.1 spec and the bootstrap registers are now recognized and displayed in clear text. They all now works with the latest release of Wireshark (1.0.5 and even with version 1.1.2 Beta).

December 9, 2008: JAI SDK Version 1.2.3 release

1) This release version was created without any changes in ver.1.2.3 Beta3.

December 5, 2008: JAI SDK Version 1.2.3 Beta 3 release

1) The control tool has improved. View windows size and location are now stored on disk for each camera every time acquisition is started and stopped. Zoom and pan in the image have been added. A new “Zoom Navigation” window is automatically displayed if the functionality is available. This will only work if the Live Video Window display is being “stretched”.
2) Click-through has been implemented for the “Start Acquisition”, “Stop Acquisition”, “Snap” and “Save” buttons. This means that the buttons will work properly even if the Main Control Tool window is not having input focus.
3) Some new features added in the .NET wrapper. E.g. Calculation of an Inter-Packet Delay Estimate added to CCamera (CCamera.EstimatedInterPacketDelay). This is a powerful tool to help adjust the Packet Delay value in systems where there are multiple GigE Vision cameras attached to the same physical network through a switch.
4) Get the last command sent in a running Script in CCamera (CCamera.GetScriptLastCommand()) in order to make the GUI better.

December 3, 2008: JAI SDK Version 1.2.3 Beta 2 release

1) Filter driver has improved not to stop showing images under the condition: XP32 or 64, RMC-6740GE, Stretched and maximized view windows.
2) The Control Tool is now checking the values we get from the cameras and that we try to send to the cameras and if the Integer or Float values are outside the legal Min/Max range or if the Integer value we try to write is not following the Increments then we log a “Warning” to the error log.
3) If any errors or warnings have been added to the error log then a small icon will appear in the lower left corner. This Icon will either be a “Warning” or an “Error” icon depending on the “maximum” level. To remove the icon again, the user need to double-click on the status-bar (or the Icon) and this will open the error log window. Inside the error log the user is now able to “acknowledge” the different errors and warnings and if all of them has been acknowledged then the icon will disappear in the Control Tool main window
4) We found a bug that made it impossible to use the latest Dalsa and Basler cameras if the PC is fast. This problem occurred at the latest plug-fest and was caused by lack of synchronization between start of stream thread and sending “AcquisitionStart” command and only happens if the camera locks “AcquisitionStart” using “TLParamsLocked”.
5) Manifest added to the Persistent IP Configuration Tool and also updated the version number in the control tool.

November 3, 2008: JAI SDK Version 1.2.3 Beta 1 release

1) Easy use of “child window” added to Jai_FactoryDotNET.dll in CCamera.StartImageAcquisition()
2) New C# sample added to demonstrate new “child window” interface.
3) We have Improved GUI of the control tool to generate software trigger, which was changed from an edit box into a button.

October 23, 2008: JAI SDK Version 1.2.2 release

1) The Control Tool manual has been translated into Japanese and added to the installers

October 24, 2008: JAI SDK Version 1.2.2 RC 1 release

1) The version of the Installer has changed; Beta1 => RC1.

October 17, 2008: JAI SDK Version 1.2.2 Beta 1 release

1) Filter driver was updated to improve its performance with Broadcom network.
2) Changed the resending-mechanism of filter Driver.
3) Changed resending source port number to same number of GVCP source port.(Filter Driver, Socket Driver, Transport Layer)
4) Updated Filter Driver, Socket Driver and Transport layer for the TM-Series camera.
5) Supporting TLParamsLocked before starting acquisition

September 25, 2008: JAI SDK Version 1.2.1 release

1) Filter driver was updated to improve its performance with Broadcom network.
2) Updated the control handling XML File of CV-M9GE
3) GenICam version 1.1.1 RC1 included in the installation

September 16, 2008: JAI SDK Version 1.2.1 Beta 8 release

1) Updated the filter driver to fix the problem that image stream stopped. (with RM-6740GE)
2) Updated the control tool not to make an error with multi ROI
3) Updated the control tool to be able to save status of "Enable packet-resend".
4) Updated the C# sample no to make build error regarding manifest

August 21, 2008: JAI SDK Version 1.2.1 Beta 7 release