Upgrade Notes

This topic lists noteworthy changes that can cause compile-time or runtime issues when upgrading Ensenso SDK or uEye driver versions in an existing application.

See this page for a list of known issues in the current version of the SDK.

Upgrade of Ensenso SDK

≥ 4.1.x

- The default stereo matching method changed to `PatchMatch2`.
- The new ImageEnhancement feature is enabled by default for monocular cameras and for stereo camera texture images.
- ComputeTexture: When texture mode is enabled, the `ComputeTexture` command now rectifies raw texture images instead of computing a texture image from rectified images.
- RenderView: The default point cloud RenderingMode was changed from `Points` to `Circles`.

B- and C-Series Firmware

- Require an update to firmware version 1.4.
- Changes to the handling of gain and blacklevel can affect the average image brightness. Review your target brightness and exposure/gain settings after updating the firmware.

≥ 4.0.x

System Requirements

- Using CUDA now requires a device with compute capability 5.0 or newer.
- Using CUDA now requires at least version 452.39 of the Nvidia driver on Windows.
- Support for Windows 7 was removed.

Default Parameters

The following parameters can change behavior of the API in the new version if you don’t set them explicitly.

- The default value of CorrespondenceThreshold and SpeckleRemoval/RegionSize changed to the same values that were already used in NxView parameter presets.
- The parameter Filling/RegionSize is now disabled by default, since it doesn’t have much effect when using the default PatchMatch stereo matching algorithm. It is still enabled in NxView parameter presets using other matching methods.
- The BandwidthLimit/Automatic parameter for the Open command now defaults to true.
- For cameras with color image sensors WhiteBalance/Automatic is now enabled by default.
- Auto exposure now uses some amount of gain before increasing the exposure time to more than 5ms for X- and XR-series cameras. This is the same behavior that was already used for C-series and newer N-series cameras.
- The default value of Gain and Automatic changed for S-series cameras. Gain now starts at the normal default value of 1 and gets adjusted automatically. Auto exposure uses some gain before increasing the exposure time.
- Virtual cameras now have FlexView enabled by default to assimilate their default parameters to those of hardware cameras.
- File cameras now preserve the position from which the data was originally captured by default. This can change the position of their data. See this topic for details.

Functional Changes

- The ImageBuffer properties for stereo cameras are changed to refer to entire image sets. [#5118] (00204276)

- The command EthernetConfiguration enables DHCP on devices that support DHCP and are connected to a host adapter that has obtained its IP address via DHCP when using the *Auto* method. [#3107]
- The command ReprojectPoints now returns right rays that originate in the right camera and intersect the left rays in the corresponding 3D points.
- The command GenerateCalibrationPattern reports an error if an unknown format is specified through the file extension of the Filename parameter instead of generating a PDF file with the wrong extension.
- The Time node for commands was moved outside of the command's result node. There is no link for backwards compatibility, accessing the node at its old location will fail. This resolves some edge cases where the Time node could overwrite command results.
  If you use the C++ API, you can replace `command.result()[itmTime]` with `command.status()[itmTime]`.
- The camera matrix node of a monocular camera now always contains the camera matrix for the full image. The actually used camera matrix with modifications due to binning can be found underneath the new Dynamic node. [#5210]
- NxLib: The PartFinder command has a new parameter TexturePenalty which changes the matching behaviour compared to previous versions. Set TexturePenalty equal to Texture to obtain the behaviour of previous versions. [#5233]
- NxLib: Bugfixes in the 2D edge layer of PartFinder will lead to improved, but different results, even with identical parameters, when Texture is enabled. [#5220]
- NxLib: Presets for PartFinder updated. `modelGeneration/medium` now uses Texture. `search/medium` now uses RefineCombined. Lowered Score for all search presets. [#5220]

Removed Functionality

- The command `ClearImages` was removed. If you used it to load images into an existing camera node, please use file cameras instead.
- The parameter `Operation` for the `Capture` and `Retrieve` commands was removed.
- Compatibility nodes for an old version of the profiler were removed from the Debug node.
- Unused string constants with no valid use cases were removed from the C headers and other interfaces.

▶ List of Removed String Constants

Changes to JSON API Functions

- The set of forbidden characters in JSON node names has been extended with the space and the double quote character as well as further special characters. See the full list here.
- `nxLibGetInt` now clips the value to the minimum or maximum possible `int` value if the underlying `double` value is outside of the range of an `int`.
- `nxLibGetBinary` now returns `NxLibBufferTooSmall` if the provided buffer is too small.
- `nxLibGetName` now returns `""` if the item is the root node.

JSON API Return Codes

- Introduced a new return code `NxLibItemPathInvalid`, which is returned by all API functions when the specified path to the item is invalid.
- Introduced a new return code `ǸxLibInvalidParameters`, which is returned by all API functions when a parameters given by the user is invalid, e.g. when passing a `NULL` pointer as `NXLIBSTR`.
- The return code `NxLibCannotCreateItem` was replaced by `NxLibItemProtected` to better describe its meaning. The old constant still exists as an alias.

The return codes of the API were adjusted in some cases to describe errors better and make return codes consistent between NxLib and NxLibRemote. These changes will probably not affect your program unless you check for specific error codes.

▶ Complete List of Changed Return Codes

C/C++ Headers

- C++ headers now require C++11. The compatibility mode for older versions using the `NXLIB_CPP_COMPATIBILITY_MODE` macro was removed. Note that the library is still binary compatible, so you can continue to compile using the C++ classes from an older version of the NxLib as a workaround.
- NxLib: The constant headers were restructured. The following unintended use cases no longer work:
    - Including the headers `nxLibApiErrors.h`, `nxLibCommonNodeNames.h` or `nxLibSymbols.h` directly.
    - Using the macros `NXLIB_ITEM`, `NXLIB_VALUE`, `NXLIB_COMMAND`, `NXLIB_ERROR`, `NXLIB_INT_CONST` or `NXLIB_APIERROR` to define additional constants outside of the NxLib header. The macros are now undefined after usage in the NxLib header.
- `NxLibItem::as<char const*>()` was removed because the returned pointer was unsafe to use under certain circumstances.

.NET Interface

- The deprecated Common Language Infrastructure (CLI) interface no longer ships with the Ensenso SDK. Use the new .NET interface instead.
  As described in the deprecation notice, the C API is still binary compatible, so you can continue to use the DLL files from older versions except that they don't contain new string constants.
- The instantiated template functions `NxLib.SetBinaryByte()`, `NxLib.SetBinaryShort()` and `NxLib.SetBinaryFloat()` have been removed. Use `NxLib.SetBinary<T>()` instead.
- The instantiated template functions `NxLib.SetBinaryFormattedByte()`, `NxLib.SetBinaryFormattedShort()` and `NxLib.SetBinaryFormattedFloat()` have been removed. Use `NxLib.SetBinaryFormatted<T>()` instead.

≥ 3.6.x

- The Common Language Infrastructure (CLI) interface is now deprecated and will be removed in a future version of the SDK. Use the new .NET interface instead, which provides a backwards compatible C# API that runs on all platforms where .NET is available.
  To simplify transition to the new interface, the Windows installers will continue to install the CLI DLL files. They are frozen to version 3.5.1394, though. Due to the stable binary interface of the NxLib these DLLs will continue to work for the forseeable future, except that new string constants and API functions will not be available. If you want to use new NxLib functionality with the CLI interface, you can use string literals or define the necessary constants for the new node names and values yourself.
- Instead of using the NxLib internal default values NxView now always loads the 'PatchMatchFlexView4' preset (or 'PatchMatchSingleShot' for cameras without FlexView) when no local settings file is available or 'Load cached settings' is unchecked.
- The CreateCamera command enforces stricter rules on the serial number of newly created file and virtual cameras. They are now limited to alphanumeric characters and the symbols `.,-+_~`.
- Improvements in PartFinder might require adjustments of score thresholds.

≥ 3.5.x

- Changes to the nxLibGetJson API function:
    - The parameter `scientificNumberFormat` was renamed to `allowScientificFormat`.
    - When `allowScientificFormat` is false, the function will never output numbers in scientific format even if they require more characters than specified by the `numberPrecision` parameter.
    - When `allowScientificFormat` is true, the function will still output numbers in non-scientific format if it is shorter than the output in scientific format would be.
    - Note that the function is binary compatible to the old version, but might return different results now.
- Initialization of the OpenGL context in the NxLib on Linux now uses EGL instead of X11. This allows rendering without running an X server, but might require some changes on headless systems. See this guide for more information.

- On Windows, the NxLib uses a dynamic C runtime library (CRT) now and requires an installation of the Microsoft Visual C++ Redistributable. The installer will install this automatically for you. You can use the NxLib DLL on a system where the installer did not run if you install the CRT manually.
- The profiler now uses a different binary format. Log files generated by an NxLib with version ≥ 3.5 cannot be opened in earlier versions of the NxProfiler.
- PartFinder's filtering of the points sampled from the CAD model and used for coverage estimation is relaxed, so an estimated relative coverage may have significantly lower values. It is recommended that you adjust the thresholds for the combined score and the coverage score.

≥ 3.3.x

- This version ships with version 2.9 of the IDS GigE Vision firmware. Any GigE Vision devices should be updated as suggested by NxView.
- Functions that require a separate license (e.g. PartFinder) now require the presence of `NxLibCmBridge64.dll` (Windows) or `libNxLibCmBridge64.so` (Linux), which gets loaded dynamically. The installers automatically install the library in the system's library directory. You don't have to do anything if you use NxLib from there.
- Due to a large number of internal changes and improvements the PartFinder command will yield different results compared to earlier versions. It is recommended to verify parametrization and results again when using PartFinder.

≥ 3.2.x

- This version adds support for EEPROM on IDS GigE Vision cameras instead of saving it to a local file (requires at least firmware 2.4). For backwards compatibility, the camera will still use the local file if it exists. Remove it manually to switch to the EEPROM on the device. Note that the SDK only ships the new firmware 2.8 for the GV-5040ACP-M, which is used in S-series cameras. For other camera model this version still ships an older firmware version due to known problems with the new firmware versions.
- This versions adds the new EEPROM calibration format 11/12. This format is only required for S-series cameras, the calibration of any other stereo camera can always be downgraded and saved in the previous format.
- The `IgnoreEnsensoPatternEncoding` parameter for CollectPattern was removed, because the resulting undecoded patterns were not usable for position estimation. CollectPattern in projector mode now always performs a basic decoding of Ensenso patterns.
- The `AbsoluteBlackLevelOffset` node in the camera parameters was removed, because it was not used and its value was always zero.

≥ 3.1.x

- Handling of multi-exposure images in the correlation matcher has been improved to increase data coverage for extreme lighting situations. This can change the matching result with these settings (SDK-2139) [062688].
- `NxLibItem::as<char const*>()` could return a dangling pointer in previous versions of the NxLib. To make it slightly safer, it now returns a pointer to which the C-API string usage restrictions apply. The method will be removed in the next major version. If you use it, you should replace it with `NxLibItem::asString()` or `NxLibItem::as<std::string>()`.
- The bugfix SDK-2351 for RenderPointMap changed the shadowing behaviour of the projection. Please verify that your application is not affected when using this command.

≥ 3.0.x

- The new PatchMatch stereo matching method as been adopted as default method and therefore the default values for the following parameters have been changed: FlexView, stereo matching Method, UniquenessRatio, TargetBrightness, ShadowingThreshold.
- In order to benefit from the unlimited measurement range of PatchMatch the new flag FullWidth has been introduced which always enforces rectified images to have maximal width, regardless of the width of the stereo matching AOI.

- The RenderPointMap parameters Near and Far did not work correctly in previous releases. If these are in use in your application please verify that they still produce the intended result with the repaired behaviour.