

areaDetector

What's New and What's Next

Mark Rivers

GeoSoilEnviroCARS, Advanced Photon Source

University of Chicago

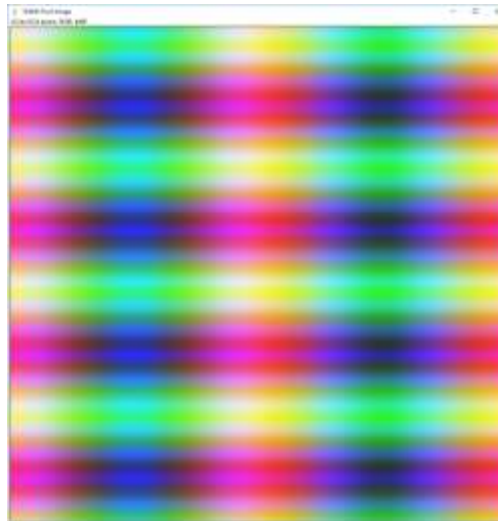
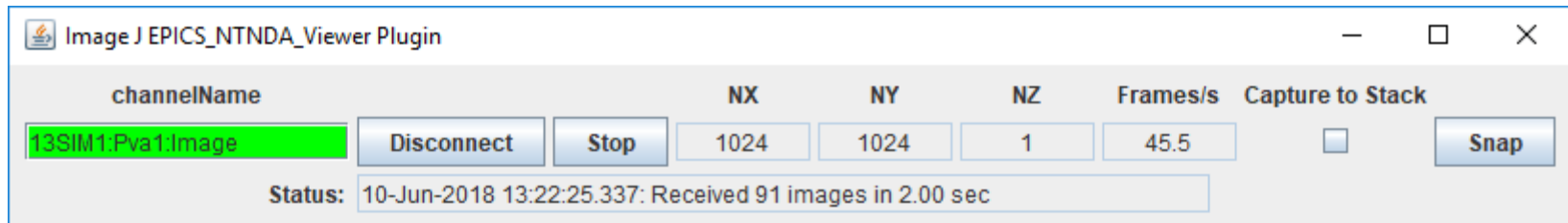


Outline

- Last update was October 2017 at ITER and ICALEPCS in Barcelona
 - areaDetector and ADCore Releases since then: R3-2, R3-3 (soon)
- Brief recap of the top items from R3-0 for those who missed it
- Major new features in 3-2 and 3-3
- Roadmap for R4-0 and R5-0

EPICS_NTNDA_Viewer ImageJ plugin (ADViewers R1-0)

- New ImageJ plugin written by Tim Madden and Marty Kraimer
- Essentially identical to EPICS_AD_Viewer.java except that it displays NTNDAArrays from the NDPluginPva plugin, i.e. using pvAccess to transport the images rather than NDPluginStdArrays which uses Channel Access.



EPICS_NTNDA_Viewer Advantages

- NTNDArray data transmitted "atomically" over the network
- With Channel Access size and data type of waveform record is fixed at iocInit, cannot be changed at runtime.
 - To view both 8-bit and 64-bit double FFT images waveform record needs to be 64-bit double, 8X network overhead for 8-bit. pvAccess changes the data type of the NTNDAArrays dynamically at run-time.
- Channel Access requires setting EPICS_CA_MAX_ARRAY_BYTES, considerable confusion and frustration for users.
- NDPluginPva is 5X-10X faster than NDPluginStdArrays
- ImageJ is 1.5-2X faster with pvAccess than with Channel Access.

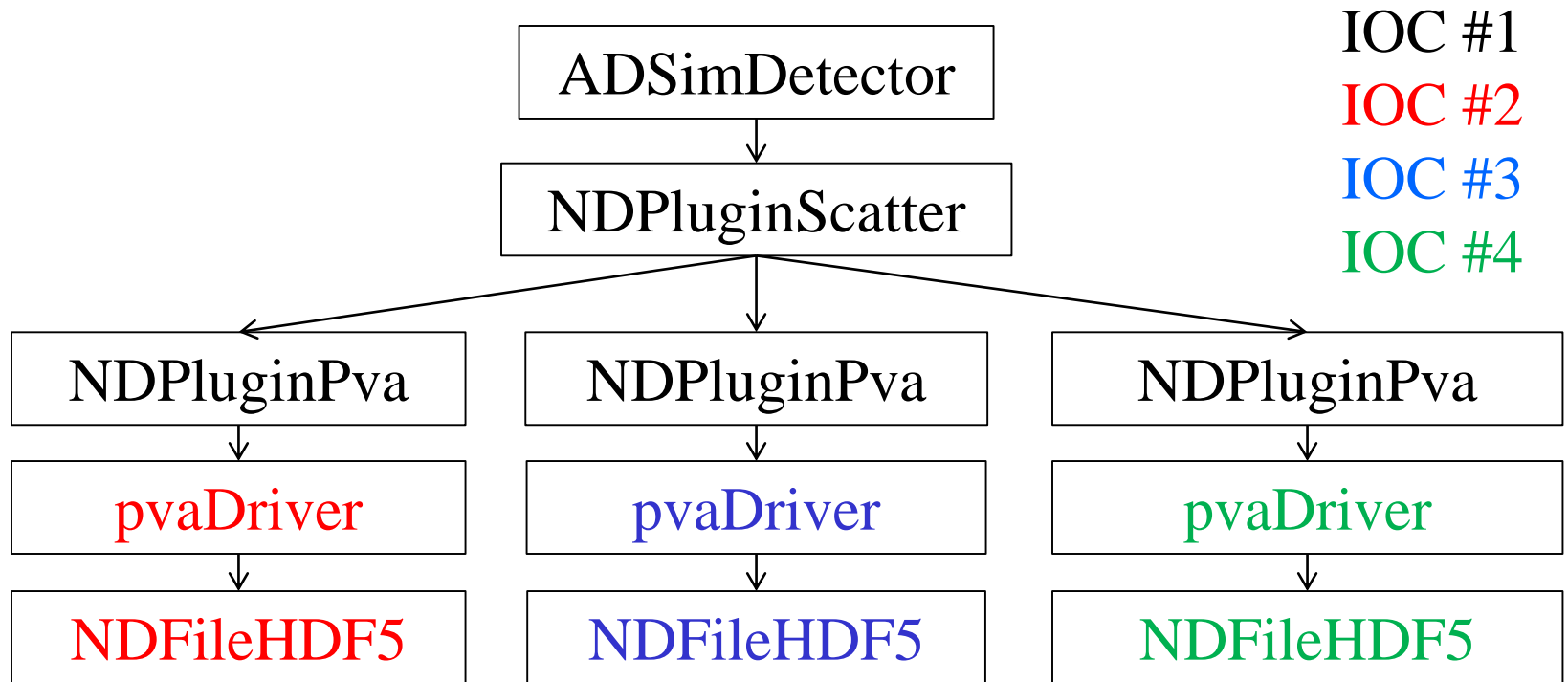
NDPluginDriver (R3-0)

Multiple Threads per Plugin

- Added support for multiple threads running the processCallbacks() function in a single plugin.
- Can improve the performance of the plugin by a large factor. Linear scaling with up to 5 threads (the largest value tested) observed for most of the plugins that now support multiple threads.
- Maximum number of threads that can be used for the plugin is set in constructor and in IOC startup script.
- Actual number of threads to use controlled via an EPICS PV at run time, up to the maximum value passed to the constructor.
- Note that plugins need to be modified to be thread-safe for multiple threads running in a single plugin object.

NDPluginScatter + V4

- Distribute file writing to multiple IOCs (4096 x 3078 8-bit)



# IOCs	Files/sec	GB/sec
1	101.0	1.19
2	195.2	2.29
3	217.5	2.55

NDArrayPool Design Enhancements (R3-3)

- Previously each plugin used its own NDArrayPool.
 - Problem: not really possible to enforce the maxMemory limits for the driver and plugin chain.
 - Sum of the memory use by the driver and all plugins that matters, not the use by each individual driver and plugin.
- NDPluginDriver base class changed to set its pNDArrayPool pointer to the address passed to it in the NDArray.pNDArrayPool for the NDArray in the callback.
- Ultimately all NDArrays are derived from the driver, either directly, or via the NDArrayPool.copy() or NDArrayPool.convert() methods.
 - Thus plugins now allocate NDArrays from the driver's NDArrayPool, not their own.

NDArryPool Design Enhancements

- maxMemory argument to the driver constructor now controls the total amount of memory that can be allocated for the driver and all downstream plugins.
- maxBuffers argument to all driver and plugin constructors is now ignored.
 - Now no limit on the number of NDArrys, only on the total amount of memory.
 - maxBuffers argument to the ADDriver and NDPluginDriver base class constructors are still present so existing drivers and plugins will work with no changes. This argument is simply ignored.
 - maxMemory argument to the NDPluginDriver constructor is only used for NDArrys allocated before the first callback, so it can safely be set to 0 (unlimited).
- These changes are generally backwards compatible.
 - However, startup scripts that set a non-zero value for maxMemory in the driver may need to increase this value because all NDArrys are now allocated from this NDArryPool.

Active Plugin Counting and Waiting for Plugins to Complete (R3-3)

- Previously to wait for plugins to complete before the driver indicated that acquisition was complete then needed to set CallbacksBlock=Yes for each plugin in the chain.
- Waiting for plugins is needed in cases like the following
 - Doing a step scan and one of the counters for the step-scan is a PV from the statistics plugin.
 - Necessary to wait for the statistics plugin to complete to be sure the PV value is for current NDAarray and not the previous one.
 - Doing a scan and writing the NDAarrays to a file with one of the file plugins. N
 - Necessary to wait for the file plugin to complete before changing the file name for the next point.
- Problems with setting CallbacksBlock=Yes.
 - Slows down the driver because the plugin is executing in the driver thread and not in its own thread.
 - Complicated to change all of the required plugin settings from CallbacksBlock=No to CallbacksBlock=Yes.

Active Plugin Counting and Waiting for Plugins to Complete

- NDPluginDriver base class now does the following:
 - Increments a NumActivePlugins counter in the driver that owns each NDArray as it is queued
 - Decrements the counter after the processing is done.
- All drivers have 3 new records:
 - **NumActivePlugins**: Indicates the total number of NDArrays that are currently processing or are queued for processing by this driver.
 - **WaitForPlugins**: Determines whether AcquireBusy waits for NumActivePlugins to go to 0 before changing to 0 when acquisition completes.
 - **AcquireBusy**: “busy” record that is set to 1 when Acquire changes to 1. It changes back to 0 when acquisition completes, i.e. when Acquire_RB V=0.
 - If WaitForPlugins is Yes then it also waits for NumActivePlugins to go to 0 before changing to 0.
- Should now rarely be necessary to change plugins to use CallbacksBlock=Yes.

simDetector.adl@corvette

Simulation Detector - 13SIM1:cam1:

Setup

asyn port SIM1
 EPICS name 13SIM1:cam1:
 Manufacturer Simulated detector
 Model Basic simulator
 Serial number No serial number
 Firmware version No firmware
 SDK version 2.7.0
 Driver version 2.7.0
 ADCore version 3.3.0

Connected

Connection
 Debugging

Shutter

Shutter mode
 Status: Det. EPICS
 Open/Close
 Delay: Open Close
 EPICS shutter setup

Plugins

Collect

Exposure time
 Acquire period
 # Images
 # Images complete 840
 # Exp./image
 Image mode Continuous
 Trigger mode Internal

Done

Acquire

Readout

	X	Y
Sensor size	1024	1024
Binning	<input type="text" value="1"/>	<input type="text" value="1"/>
Region start	<input type="text" value="0"/>	<input type="text" value="0"/>
Region size	<input type="text" value="1024"/>	<input type="text" value="1024"/>
Reverse	<input type="text" value="No"/>	<input type="text" value="No"/>
Image size	1024	1024
Image size (bytes)	1048576	1048576
Gain	<input type="text" value="1.000"/>	1.000
Data type	<input type="text" value="UInt8"/>	UInt8
Color mode	<input type="text" value="Mono"/>	Mono

Simulation setup

active plugins 606
 Wait for plugins
 Acquire busy Acquiring

Detector state
 Time remaining
 Image counter 2876
 Image rate
 Array callbacks Enable

Attributes

File
 Macros
 Status Attributes file OK

Buffers

Buffers used 638
 Buffers alloc/free 1035 397
 Memory max/used (MB) 0.0 1591.0
 Buffer & memory polling

← New records

NDFileTIFF (R3-3)

- Added support for readFile() so it is now possible to read a TIFF file into an NDArray using this plugin and do callbacks to downstream plugins.
 - All datatypes (NDDataType_t) are supported.
 - Supports Mono, RGB1, and RGB3 color modes. It also correctly reads files written with RGB2 color mode.
 - Stores the NDArray fields uniqueID, timeStamp, and epicsTS if they are present.
 - Restores all of the NDArray NDAttributes that were written to the TIFF file.
 - Because of the way the NDAttributes are stored in the TIFF file the restored attributes are all of type NDAttrString, rather than the numeric data types the attributes may have originally used.
- One motivation is for NDPluginProcess to be able to read TIFF files for the background and flat field images, rather than needing to collect them each time it is used.

NDPluginProcess (R3-3)

- Load a dedicated TIFF plugin for the NDPluginProcess plugin in commonPlugins.cmd.
 - Used for reading background or flatfield TIFF files.
- Add an sseq record to load the background image from a TIFF file. Executes all the following steps:
 1. Saves the current NDArrayPort fo the Process plugin to a temporary location
 2. Sets the NDArrayPort to the TIFF plugin.
 3. Enables ArrayCallbacks for the TIFF plugin in case they were disabled.
 4. Process the ReadFile record in the TIFF plugin. This reads the TIFF file and does callback to the Process plugin.
 5. Loads the NDArray from the callback into the background image.
 6. Restores the previous NDArrayPort from the temporary location.
- Add an sseq record to load the flatfile from a TIFF file.
 - Executes the same steps as for the background above, except that in step 5 it loads the NDArray into the flatfile image.

NDPluginProcess (R3-3)

New records

The screenshot shows the NDPluginProcess control panel for 13SIM1:Proc1. The interface is divided into several sections:

- Left Panel:** Configuration for the plugin, including asyn port (PROC1), plugin type (NDPluginProcess), ADCore version (3.2.0), plugin version (3.2.0), array port (SIM1), array address (0), enable (checked), min. time (0.000), queue size/free (0000), array counter (Reset to 0 | 13), array rate (0.00), execution time (5.553 msec), dropped arrays (Reset to 0 | 0), # dimensions (2), array size (1024 1024 0), data type (UInt3), color mode (Mono), unique ID (11), time stamp (680889742.798), array callbacks (checked), and process plugin (Process).
- Background subtraction:** Save background (Save | Val5d), enable background (Disable | 00000), and Read TIFF file (Read | QQuit).
- Flat field normalization:** Save flat field (Save | Val5d), enable flat field (Disable | checked), scale flat field (0.00 | 0.00), and Read TIFF file (Read | QQuit).
- Scale and Offset:** Enable scale/off. (Disable | checked), auto scale/off. (Auto calc), scale value (1.00 | 1.00), and offset value (0.00 | 0.00).
- Low/High Clipping:** Enable low clip (Disable | checked), low clip value (0 | 0), enable high clip (Disable | checked), and high clip value (000 | 100).
- Output data type:** Data type (Automatic | Automatic).
- Recursive filter:** Enable filter (Disable | checked), N filter (1), N filtered (0), filter type (Recursive), reset filter (Reset), auto reset filter (No), filter callbacks (Every array), and various offset and scale parameters (DOffset, OScale, OC1, OC2, OC3, OC4, FOffset, FScale, FC1, FC2, FC3, FC4, ROffset, RC1, RC2).

Red arrows point from the text "New records" to the "Read TIFF file" buttons in the Background subtraction and Flat field normalization sections.

The screenshot shows the NDProcessTIFF control panel for 13SIM1:Proc1:TIFF. The interface is divided into two main sections:

- Left Panel:** Configuration for the plugin, including asyn port (PROC1TIFF), plugin type (NDFileTIFF), ADCore version (3.2.0), plugin version (3.2.0), array port (SIM1), array address (0), enable (checked), min. time (0.000), queue size/free (00), array counter (Reset to 0 | 2), array rate (0.00), execution time (0.000 msec), dropped arrays (Reset to 0 | 0), # dimensions (2), array size (1024 1024 0), data type (UInt3), color mode (Mono), unique ID (1), time stamp (68073589.798), array callbacks (checked), and process plugin (Process).
- Right Panel:** File path (/home/epics/scratch), File name (test_flat), Next file # (1 | 1), Auto increment (No | No), Filename format (ts%_X3.3d.tif, Example: ts%_X3.3d.(EXT)), Last filename (/home/epics/scratch/test_flat_001.tif), Read status (Write OK), and Read message.

NDFileHDF5 (R3-2)

- Added support for blosc compression library.
 - Compressors include blosclz, lz4, lz4hc, snappy, zlib, and zstd.
 - Also support for ByteSuffle and BitShuffle.
 - ADSupport now contains the blosc library, so it is available for most architectures.
 - The build flags WITH_BLOSC, BLOSC_EXTERNAL, and BLOSC_LIB have been added, similar to other optional libraries.
Thanks to Xiaoqiang Wang for this addition.
- Changed all output records in NDFileHDF.template to have PINI=YES. This is how other plugins all work.

NDPluginFile Capture Mode (R3-2)

- Changed the way that capture mode is implemented.
- Previously it created NumCapture NDArrays using the "new" operator. As NDArrays arrived it copied the data and attributes into these arrays. This had several problems:
 - NDArrays were not allocated from the NDArrayPool, so memory limits were not enforced.
 - Because not allocated from NDArrayPool if passed to other functions or plugins there would be problems with attempts to call NDArray::reserve() or release().
 - The copy operation is inefficient and not a good idea.
- Changed to:
 - Allocate an array of NumCapture pointers.
 - As NDArrays arrive their reference count is incremented with reserve().
 - NDArray address is copied to the pointer array
 - After the files are written the cleanup routine now simply decrements the reference counter with release(), rather than having to delete the NDArray.
 - New scheme is much cleaner.
 - May require some changes to startup scripts to increase NDArrayPool size.

NDPluginOverlay (R3-2)

- Improved behavior when changing the size of an overlay.
 - Previously the Position was always preserved when the Size was changed.
 - Not the desired behavior when the user had set the Center rather than Position.
 - Code now remembers whether Position or Center was last set, and preserves the appropriate value when the Size is changed.
- Overlays were previously constrained to fit inside the image on $X=0$ and $Y=0$ edges.
 - User may want part of the overlay to be outside the image area.
 - Location of the overlay can now be set anywhere, including negative positions.
 - Each pixel in the overlay is now only added if it is within the image area.
- Changed the cross overlay so that it is drawn symmetrically with the same number of pixels on each side of the center.
 - Actual size is $2 * \text{Size} / 2 + 1$, $= \text{Size} + 1$ if Size is even.

Operator displays

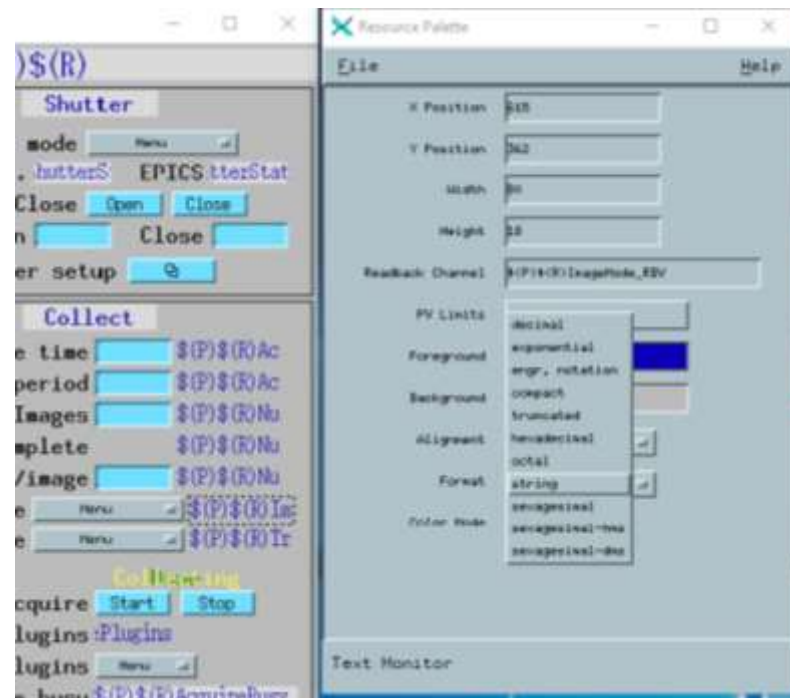
medm, edm, caQtDM, CSS-BOY (R3-2)

- Added ADApp/op/Makefile.
 - Runs the conversion tools to convert the medm adl files to edl for edm, ui for caQtDM, and opi for CSS-BOY.
- RULES_OPI file was added to synApps/support/configure to support this.
- If RULES_OPI file is not found the Makefile does nothing.
- If the RULES_OPI file is found then a CONFIG_SITE file in synApps/configure or in EPICS base must define these symbols:
 - **ADL2EDL**: path to adl2edl for edm
 - **ADL2UI**: path to adl2ui for caQtDM
 - **CSS**: path to css. Must be a recent version that supports the command
`css -nosplash -application org.csstudio.opibuilder.adl2boy.application`
- edl/autoconvert, ui/autoconvert, and opi/autoconvert directories contain new conversions of all of the medm files.
- edl, ui, and opi directories should contain only manually tweaked files.
 - Many of the files in these directories have been removed, either because they were actually old autoconverted files, or because they are obsolete and the new autoconverted files are better.

medm File Improvements (R3-2, R3-3)

- Good conversion requires a good medm adl file
- Text graphics widget sizes are set to the actual size of the text.
 - medm will display text outside the widget if it is not large enough, but other display managers will not.
- Text update widgets set to the correct datatype.
 - medm will display an enum widget as a string even if the datatype is set to "decimal" rather than "string", but other display managers will not.

```
Exposure time  $(P)$ (R) Ac
Acquire period  $(P)$ (R) Ac
# Images  $(P)$ (R) Nu
# Images complete  $(P)$ (R) Nu
# Exp./image  $(P)$ (R) Nu
```



medm

Simulation Detector - 13SIM1:cam1

Setup

asyn port: SIM1
EPICS name: 13SIM1:cam1
Manufacturer: Simulated detector
Model: Basic simulator
Serial number: No serial number
Firmware version: No firmware
SDK version: 2.7.0
Driver version: 2.7.0
ADCore version: 3.3.0

Connected

Connection:
Debugging:

Shutter

Shutter mode:

Status: Det. EPICS:

Open/Close:

Delay: Open: Close:

EPICS shutter setup:

Collect

Exposure time:
Acquire period:
Images:
Images complete: 401882
Exp./image:
Image mode:
Trigger mode:

Acquire:

active plugins: 1011
Wait for plugins:

Acquire busy:

Detector state:

Time remaining:

Image counter:

Image rate:

Array callbacks:

Plugins

Stats:

Readout

	X	Y
Sensor size	<input type="text" value="1024"/>	<input type="text" value="1024"/>
Binning	<input type="text" value="1"/>	<input type="text" value="1"/>
Region start	<input type="text" value="0"/>	<input type="text" value="0"/>
Region size	<input type="text" value="1024"/>	<input type="text" value="1024"/>
Reverse	<input type="button" value="No"/>	<input type="button" value="No"/>
Image size	<input type="text" value="1024"/>	<input type="text" value="1024"/>
Image size (bytes)	<input type="text" value="1048576"/>	
Gain	<input type="text" value="1.000"/>	<input type="text" value="1.000"/>
Data type	<input type="button" value="UInt8"/>	<input type="button" value="UInt8"/>
Color mode	<input type="button" value="None"/>	<input type="button" value="Mono"/>

Simulation setup:

Attributes

File:
Macros:
Status:

Buffers

Buffers used: 1018
Buffers alloc/free: 1048 30
Memory max/used (MB): 0.0 1216.0
Buffer & memory polling:

CSS-BOY

Simulation Detector - 13SIM1:cam1

Setup

asyn port: SIM1
EPICS name: 13SIM1:cam1
Manufacturer: Simulated detector
Model: Basic simulator
Serial number: No serial number
Firmware version: No firmware
SDK version: 2.7.0
Driver version: 2.7.0
ADCore version: 3.3.0

Connected

Connection:
Debugging:

Shutter

Shutter mode:

Status: Det. EPICS:

Open/Close:

Delay: Open: Close:

EPICS shutter setup:

Collect

Exposure time:
Acquire period:
Images:
Images complete: 41345
Exp./image:
Image mode:
Trigger mode:

Acquire:

active plugins: 1010
Wait for plugins:

Acquire busy:

Detector state:

Time remaining:

Image counter:

Image rate:

Array callbacks:

Plugins

Stats:

Readout

	X	Y
Sensor size	<input type="text" value="1024"/>	<input type="text" value="1024"/>
Binning	<input type="text" value="1"/>	<input type="text" value="1"/>
Region start	<input type="text" value="0"/>	<input type="text" value="0"/>
Region size	<input type="text" value="1024"/>	<input type="text" value="1024"/>
Reverse	<input type="button" value="No"/>	<input type="button" value="No"/>
Image size	<input type="text" value="1024"/>	<input type="text" value="1024"/>
Image size (bytes)	<input type="text" value="1048576"/>	
Gain	<input type="text" value="1.000"/>	<input type="text" value="1.000"/>
Data type	<input type="button" value="UInt8"/>	<input type="button" value="UInt8"/>
Color mode	<input type="button" value="Mono"/>	<input type="button" value="Mono"/>

Simulation setup:

Attributes

File:
Macros:
Status:

Buffers

Buffers used: 1018
Buffers alloc/free: 1048 30
Memory max/used (MB): 0.0 1222.0
Buffer & memory polling:

medm

Simulation Detector - 13SIM1:cam1

Setup

asyn port SIM1
EPICS name 13SIM1:cam1
Manufacturer Simulated detector
Model Basic simulator
Serial number No serial number
Firmware version No firmware
SDK version 2.7.0
Driver version 2.7.0
ADCore version 3.3.0
Connected
Connection
Debugging

Shutter

Shutter mode
Status: Det. EPICS
Open/Close
Delay: Open 0.000 Close 0.000
EPICS shutter setup

Collect

Exposure time 0.001 0.001
Acquire period 0.010 0.010
Images 5 5
Images complete 401882
Exp./image 1 1
Image mode Continuous
Trigger mode Internal

Readout

	X	Y
Sensor size	1024	1024
Binning	1	1
Region start	0	0
Region size	1024	1024
Reverse	No	No
Image size	1024	1024
Image size (bytes)	1048576	1048576
Gain	1.000	1.000
Data type	UInt8	UInt8
Color mode	None	Mono

Plugins

ALL File ROI
Stats

Attributes

File
Macros
Status

Buffers

Buffers used 1018
Buffers alloc/free 1048 30
Memory max/used (MB) 0.0 1216.0
Buffer & memory polling

caQtDM

Simulation Detector - 13SIM1:cam1

Setup

asyn port SIM1
EPICS name 13SIM1:cam1
Manufacturer Simulated detector
Model Basic simulator
Serial number No serial number
Firmware version No firmware
SDK version 2.7.0
Driver version 2.7.0
ADCore version 3.3.0
Connected
Connection
Debugging

Shutter

Shutter mode
Status: Det. EPICS
Open/Close
Delay: Open 0.000 Close 0.000
EPICS shutter-setup

Collect

Exposure time 0.001 0.001
Acquire period 0.010 0.010
Images 5 5
Images complete 65336
Exp./Image 1 1
Image mode Continuous
Trigger mode Internal

Readout

	x	y
Sensor size	1024	1024
Binning	1	1
Region start	0	0
Region size	1024	1024
Reverse	No	No
Image size	1024	1024
Image size (bytes)	1048576	1048576
Gain	1.000	1.000
Data type	UInt8	UInt8
Color mode	Mono	Mono

Plugins

ALL File ROI
Stats

Attributes

File
Macros
Status

Buffers

Buffers used 1020
Buffers alloc/free 1048 28
Memory max/used (MB) 0.0 1224.0
Buffer & memory polling

- caQtDM comes with a good adl to ui conversion tool, no changes needed

medm

edm

- adl2edl required changes for parallel make because it used a hardcoded temporary file name (where to push fix?)
- Quality of conversions is poor. adl2edl needs work.

Other Changes (R3-3)

- **NDArrayPool Enhancements**
 - Changes to allow inheriting it from derived classes. Thanks to Sinesa Veseli for this, he will present later this morning.
 - Optimization to memory allocation mechanism. Original work by Michael Huth. I am currently modifying to use `std::multiset`, same as used for plugin output sorting.
- **ntndArrayConverter.cpp**
 - Added conversion of the `NDArray.timeStamp` and `NDArray.epicsTS` fields from EPICS epoch (Jan. 1 1990) to Posix epoch (Jan. 1, 1970).
 - Needed because NDArrays use EPICS epoch but `pvAccess` uses Posix epoch and the timestamps shown by `pvGet` were incorrect for the NTNDArrays.

Point Grey 10-Gbit Ethernet Camera

Oryx ORX-10G-51S5C-C

- 2448 x 2048 global shutter CMOS
- Sony IMX250 2/3”
- Dynamic range of 72 dB
- Peak QE of 62%
- Read noise of 2.2e-
- 12-bit, 10-bit, or 8-bit data
- Max frame rate of 162 fps
 - 779 MB/S, >8X faster than GigE
- \$1,875



Model	Resolution	Price	Speed (frames/s)	Speed (MB/s)
ORX-10G-123S6M-C	4096x3000	\$3,950	68 frames/s	797 MB/s
ORX-10G-123S6M-C	4096 x 2160	\$3,650	93 frames/s	785 MB/s
ORX-10G-51S5M-C	2448x2048	\$1,875	163 frames/s	779 MB/s

ADSpinnaker

- New driver for Point Grey GeniCAM cameras using their Spinnaker SDK (10 GigE, GigE, USB-3)
 - Currently working on Windows
 - Linux requires Ubuntu 16 (gcc 5.4, special release of ffmpeg)
- Some work beginning on aravisGigE driver
 - Guabao Shen at APS and Neil O'Brien at Diamond)



ADPixirad

- Added support for the PIXI-III chip
- New CountMode record to select Normal, NPI and NPISUM count modes.
- New HitThreshold record to control the VtHit value.

The screenshot displays the ADPixirad software interface, titled "Pixirad Detector Control - 13PR1:cam1:". The interface is organized into several panels:

- Setup:** Contains fields for "asyn port PIXIRAD", "EPICS name 13PR1:cam1:", "Manufacturer Pixirad", "Model Pixirad-2", "Serial number 2010", "Firmware version PX2-Dec2016.1.1", "SDK version Unknown", "Driver version 2.2.0", and "ADCORE version 2.6.0". It also shows a "Connected" status and buttons for "Connect", "Disconnect", and "Debugging".
- Shutter:** Features a "Shutter mode" dropdown set to "None", "Status: Det. closed EPICS", "Open/Close" buttons, "Delay: Open 0.000 Close 0.000", and an "EPICS shutter setup" button.
- High Voltage:** Includes "HV mode" (Auto), "HV state" (Off), "HV setpoint" (300.0 V), "HV value" (0.0 V), and "HV current" (0.3 mA).
- Environmental:** Shows "Cooling" (On), "Setpoint" (-30.0 C), "Cold Temp." (-27.6 C), "Hot Temp." (27.3 C), "Box Temp." (19.7 C), "Box humidity" (1.3%), "Dew point" (-36.6 C), "Peltier power" (100.0%), and "Status".
- Attributes:** A field for "File" is present.
- Detector:** Lists "Detector Size" (402 1024), four "Threshold" values (20.0, 20.0, 30.0, 40.0 keV), "Sync in polarity" (Pos), "Sync out polarity" (Pos), "Sync out function" (Read done), "Auto calibrate" (AutoCal), and "System reset" (Reset).
- Collect:** Includes "Exposure time" (5.000), "Acquire period" (0.000), "# Images" (1), "Images collected" (1), "Colors collected" (0), "UDP buffs read" (1), "UDP speed (MB/s)" (81.8), "UDP buffs free/max" (1500/1500), "Frame type" (1 color high), "Trigger mode" (Internal), "Count mode" (Normal), "Acquire" (Start/Stop), "Image counter" (2), "Image rate" (0.00), and "Array callbacks" (Enable).
- Status:** Displays "Status: Server returned OK", "To server: DAQ:#! LOOP 1 5000 0 1COL1 INT UNMOD AUTOHV", "From server: DETECTOR 2010 GOT:DAQ:#! LOOP 1 5000 0 1COL1 INT UNMOD AUTOHV", and "System info: MOD-PIXIRAD-2 REV16-10/16/16 ASIC-PIXIR3 SER008-1024 TH_C008-10/16/16 'FIRMV101: 2214220'".

Roadmap: ADCore R4-0

- Put more functionality into ADDriver base class
 - Derived class would call `ADDriver::doPluginCallbacks()`, which would handle setting attributes, getting timestamp, calling plugins, etc.
- Simplify file saving modes (no more Single, Capture, Stream) and eliminate AutoSave
- Add flag to prevent overwriting files

Roadmap: ADCore R5-0

- Change NDArray to NTNDArray for passing data to plugins
- Use PVDatabase
 - “local” provider within IOC
 - “pva” provider between IOCs
- Smart pointers automatically eliminate all unnecessary copying
- Eliminates need for NDPluginPva
- V4 clients can immediately receive data from any point in plugin chain
- Distribute load to multiple IOCs without pvaDriver
- Bruno Martins has demonstrated this working for ADSimDetector and NDPluginStats