

JCAM

The internal raw capture app



version 2.9.1

The screenshot displays the JCAM application interface across three main tabs: **Raw Capture**, **Settings**, and **Back Camera**.

Raw Capture Tab:

- Header buttons: Settings, Raw Capture (selected), Reprocess.
- Image preview: Shows two toys, a brown bear and a green troll, sitting together.
- Metadata panel: Displays a small thumbnail of the image and a color histogram.
- Control buttons: quit, 1, 2, capture.
- Focus controls: Front, Back, focus, Start, Stop.
- Text at the bottom: 20111201_17580025_0000.

Settings Tab:

- Buttons: Raw Capture, Settings (selected), User Guide... .
- Switches:
 - Enable test pattern overlay: off
 - Mirror front camera: ON
 - Face detect assisted AE: ON
 - Auto-update color histogram: ON
- Buttons: focus mode..., Advanced Settings, Front Camera Format..., Back Camera Format..., Histograms in metadata (with OFF switch).

Back Camera Tab:

- Header buttons: Cancel, Back Camera (selected), Save.
- List of camera configurations:
 - [08] 2112x1185, 30fps, scale=0.750
 - [09] 1408x792, 30fps, scale=0.500
 - [10] 704x528, 30fps, bin, scale=0.880
 - [11] 1408x792, 60fps, bin, scale=1.000
 - [12] 1600x1200, 20fps, scale=0.490
 - [13] 2048x1536, 20fps, scale=0.627
 - [14] 2592x1936, 20fps, scale=0.794
 - [15] 848x640, 20fps, scale=0.260 ✓
 - [16] 1280x720, 60fps, scale=0.455
 - [17] 1280x720, 60fps, scale=0.909

JCAM

Jcam is an internal app for capturing raw images, JPEGs, color histogram information, and metadata. It also includes ISP analysis tools such as a powerful interactive metadata browser, and a live color histogram viewer. The raw captures help our camera engineers tune processing algorithms and diagnose scene failures.

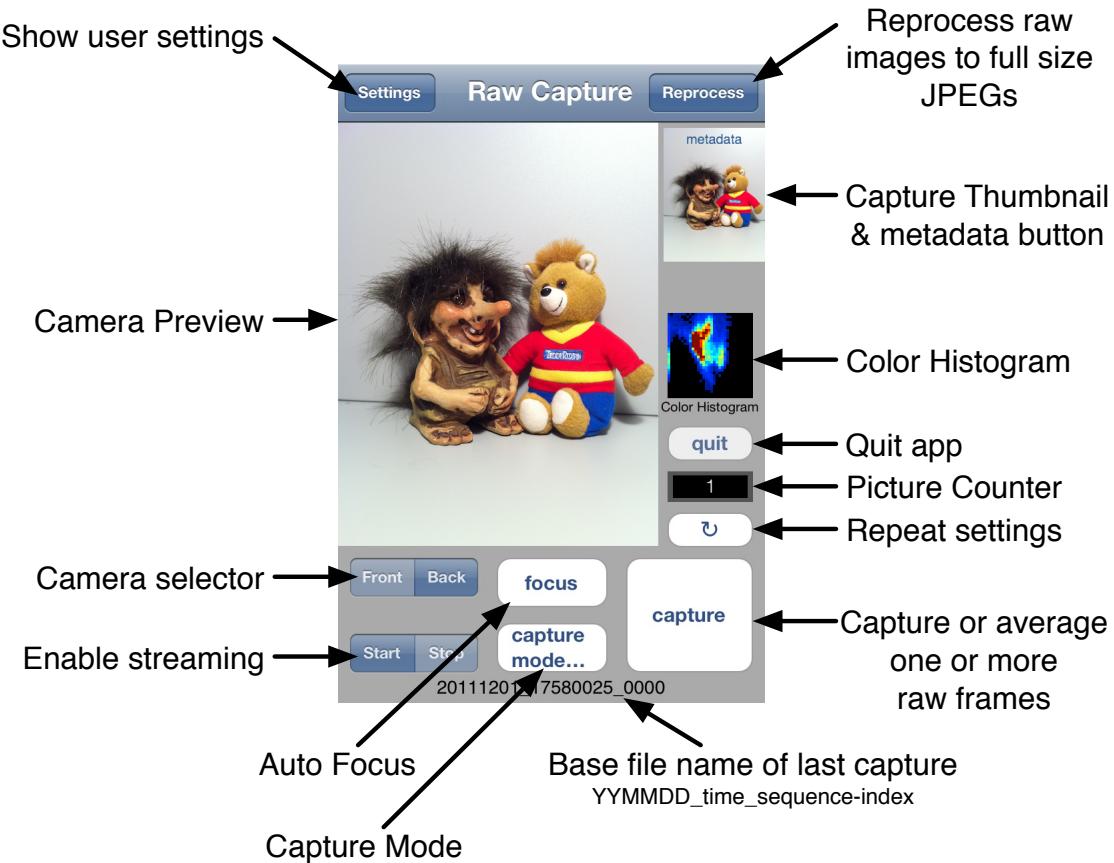
Quick Start Guide



1. Use the **Front/Back** button to select which camera to use.
2. Press the **Start/Stop** button to begin streaming.
3. The **focus** button will issue an autofocus command, if the camera supports it.
4. Press the **Capture** button to capture the current image. It is stored in `/var/mobile/Documents/jcam/`. Stored images have a suffix of `.raw` for the Bayer data, `.meta` for metadata, `.JPEG` for a jpeg representation, and `.colorHist` for color histogram data.
5. After taking pictures, the **quit** button provides a fast way to exit the app.
6. It is easy to retrieve the captured files using rsync from Terminal:
`rsync -av rsync://root@localhost:10873/root/var/mobile/Documents/jcam/*` .
Note there is a trailing “dot” in the sample command line above (this is the rsync destination).
7. Flash capture, bracketed-capture sequences, and frame averaging are available by touching the “**capture mode...**” button. Repeat captures (for frame sequences), with an optional delay (for time-lapse recording) are available by tapping the “ \cup ” button. Additional settings are available by touching the “**Settings**” button

[To do: Screenshots still need to be updated to iOS 7...]

Raw Capture (main view)



Tip: Double tap the Camera Preview to zoom in and out [New in jcam 2.2]

The Raw Capture screen provides a simple interface to allow either camera to be selected, and for the streaming to be enabled, or paused. When streaming is enabled the camera preview is updated with each frame and the **capture** button saves the current frame. Jcam saves raw and metadata files, a color histogram, and a JPEG based on the current stream format (choosable from Settings).

Tip: The side volume buttons can be used to adjust the volume of the camera shutter sound. To enable this, use the "Settings" app, and under "Sounds" turn on the button labeled "Change with Buttons".

The **Picture Counter** starts at zero each time the app is started, and increments with each capture. When two units with different cameras are used for side-by-side comparisons of many different capture scenarios, this counter can be useful to ensure that the same number of pictures have been taken with each camera, so the side-by-side captures remain in sync.

The bottom **Status Area** shows the “base file name” of the most recent capture. This includes the date (YYYYMMDD), the time in hours, minutes, seconds, and hundredths of a second, and a sequence index number (formerly the device serial number). Those who are paranoid, may think of this as a way to track their every movement, but it is just a naming convention to provide unique file names and a method to correlate which raw images came from which device for tuning and troubleshooting.

The **Color Histogram** is used for tuning and development of the Auto White Balance algorithm.

The **Capture Mode...** button provides a menu to choose between a normal capture, flash capture, averaged capture, multi-frame capture, and various bracketed capture modes, including a user-configurable bracketed capture which allows up to seven user-specified relative exposure values to be specified.

The capture thumbnail view shows the last captured image, and also doubles as a **Metadata** button. Tap the thumbnail to view its metadata.

The navigation bar has a **Reprocess** button to reprocess raw images, and a **Settings** button, which allows for various settings to be changed, as described under *Settings*.

The **quit** button provides a convenient way to exit the app. At one point it was going to be removed, but jcam users voiced strong opposition to the change. Maybe it violates user interface guidelines, or perhaps it is establishing new paradigms for iOS user controls ☺

S A V E D I M A G E S

Saved images will be in `/var/mobile/Documents/jcam/`, and have file names with `date_time_sequenceIndex_widthxheight` and the following suffixes:

```
.JPEG      // 4:2:0 saved as JPEG
.meta      // the meta data
.raw       // the raw Bayer data
.colorHist // Color Histogram data
.NV12      // uncompressed “yuv” (luma plane followed by Cb/Cr plane)
```

Tip: Here is a quick way to get captured images off the units:

```
rsync -av rsync://root@localhost:10873/root/var/mobile/Documents/jcam/* .
```

Note there is a trailing “dot” in the sample command line above (this is the rsync destination).

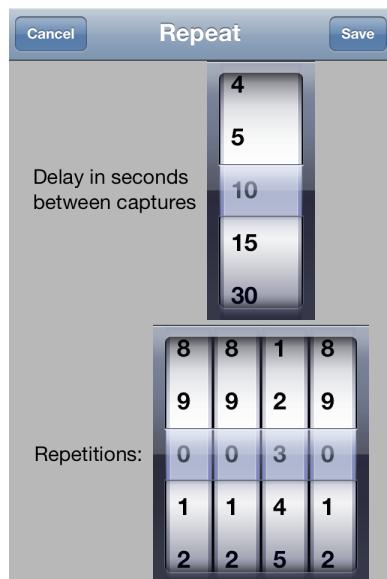
The width and height in the raw file name reflect the number of unpadded samples. However the raw data may be padded to a multiple of a power-of-two. The padded bytes-per-row can be determined by dividing the total raw size by the height.

Repeated Capture / Time-Lapse Settings



Repeat Button

The repeat button on the Raw Capture (main) view allows for a user specified repeat setting, to take multiple captures with an optional delay between each capture.

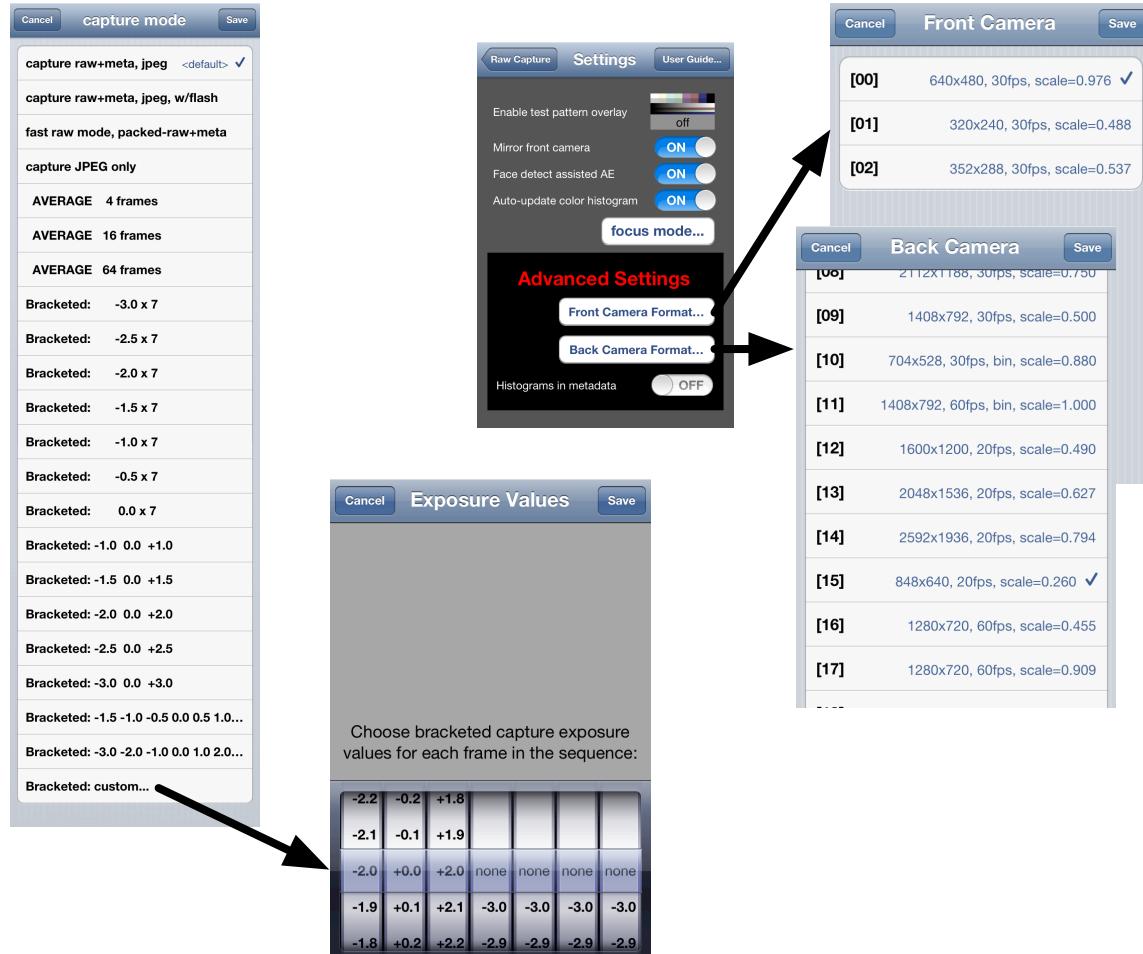


Repeat Panel

The Repeat panel allows for the specified capture operation to be repeated, with an optional delay after each capture. For example, if the capture operation is “JPEG-only”, one can capture 300 frames with zero seconds between frames, to capture a 300 frame motion jpeg sequence. Or by specifying a delay between each capture, it is easy to create a time-lapse recording. The repeat mode can be used with any capture operation, including averaged and bracketed captures.

Settings

The Settings panel provides additional control for some jcam features, as detailed below:



- The front camera preview is normally mirrored. The “**Mirror front camera**” button in jcam Settings can turn mirroring on or off.
- The “**Face detect assisted AE**” button toggles the face detect AE setting on/off.
- The color histogram can update live, or only update with each captured frame, depending on the state of the “**Auto-update color histogram**” button.
- The “**Capture Mode...**” menu was formerly in the settings panel, but is now available directly from the Raw Capture (main) view. This is initially set to “capture raw+meta, jpeg”, but it can also allow for a single frame capture with flash, or a frame sequence of multiple frames can be selected. The frame sequence does not guarantee all frames to be sequential because it may take longer than a frame time to write each frame (especially for higher resolutions). A frame sequence can also be averaged, in which case an averaged raw frame

will be saved, along with the last frame in the averaged sequence, which will also include the last frame's metadata and jpeg. Various bracketed capture modes are also available, including a user-configurable bracketed capture with up to seven user-specified relative exposure values.

- **New in jcam 2.3:** “fast raw mode, packed-raw+meta” is a special **Capture Mode** which packs each raw frame tossing unused bits to reduce the file write time (the default H3/H4/H5 RAW10 mode only has 10 bits of raw data per sample, aligned on 16-bit boundaries). This allows saving raw VGA sizes at 30 fps writing frame sequences (repeat count greater than one). Use **Reprocess** to automatically unpack the packed raws.

Saved Formats	Raw	Meta	Jpeg	Color Histogram	Nv12
raw+meta, jpeg	✓	✓	✓	✓ ¹	
fast raw capture	✓ ³	✓		✓ ¹	
flash capture	✓	✓	✓	✓ ¹	
averaged capture	✓ ²	✓ ²	✓ ²		
bracketed capture	✓	✓			
JPEG-only capture			✓		
reprocess raw			✓		✓

1. A Color Histogram is only saved for the last frame when performing repeated capture operations.
2. Averaged captures save a raw for the averaged frame and copies the metadata for the last frame as the “averaged” metadata. Additionally the (unaveraged) last frame's raw, meta, and JPEG are saved for reference.
3. Fast raw capture saves raws in a compressed format for speed. **jcam's Reprocess Raw will uncompress them.**

More advanced options are available to choose the streaming format for the front and back cameras.

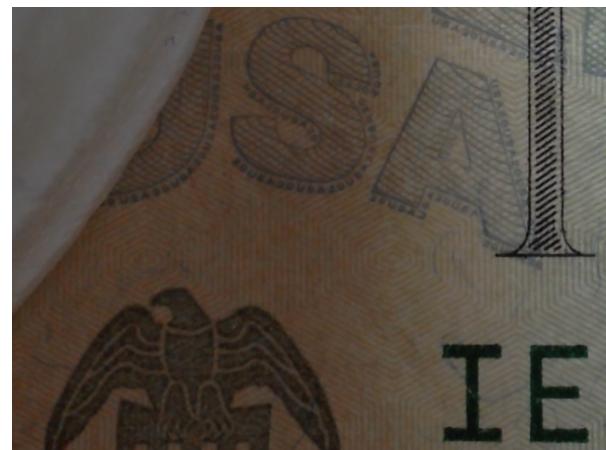
Notes:

- Bracketed captures will have a suffix of “_<bracketed_capture_sequence_name>[xx]”, where xx is the index into the sequence, and index 00 is always the pre-bracketed capture. The pre-bracketed capture has DRC enabled, while the bracketed captures do not.
- The thumbnail displayed for bracketed captures will be the pre-bracketed capture frame

- Any camera mode with a width or height greater than 2048 (or 4096 on some devices) will not display in the preview view. (These frames are passed to a Core Animation queue, without error, and it silently fails.) This is an undocumented limitation which does not have any workaround at this time.
- The **Enable Histograms** switch returns histograms in the metadata and sets the “bHistEnable” field in the “Shared Output/Histogram metadata”. There is an extra expense to enable this, so leave it off, unless you know you need it.
- **Full Statistics** metadata can be enabled with a defaults write (there is no UI for this setting). This substantially increases the metadata size, so it should only be enabled when needed. [*To enable*: login as mobile, then “defaults write com.apple.coremedia H4ISPFULLStats 1”, then “killall mediaserverd”. *To disable*: login as mobile, then “defaults delete com.apple.coremedia H4ISPFULLStats”, then “killall mediaserverd”.]
- Averaging does a good job at removing chroma noise, as shown below:



Left: Close up image from N94 (dim light)



Right: Same capture using jcam's 64 frame average setting

Metadata

jcam includes a powerful, interactive, metadata browser. Surprisingly, jcam does not know anything about the metadata. Instead it uses a metadata formatter which converts metadata structures into lists of names and values which can be displayed in the metadata view for debugging. The values may be formatted as integers, booleans ("on" & "off"), floating point, symbolic enumerations, ... or a value may be lists of names and values, and those values can also be lists of names and values, allowing for metadata to be displayed in a hierarchical view to an arbitrary depth. (Currently, only H4 metadata is supported).

The diagram illustrates the workflow of metadata in jcam. It starts with a screenshot of the jcam application interface, showing a raw capture screen with a teddy bear image and various controls like 'capture' and 'start/stop'. An arrow points from this screen to a 'Metadata' view, which displays a hierarchical list of metadata fields under a 'header' section. This list includes 'Shared Input', 'Shared Output', 'Shared Auto Exposure', 'Shared Auto Focus', 'Shared Back-End', 'Shared Front-End', 'Apple Auto Exposure', 'Apple Auto Focus', and 'Apple Auto White Balance'. Another arrow points from this view to a 'header' metadata view, which shows details like 'metadata version' (19), 'Image capture time-stamp' (399752470), 'frame counter' (102), and 'total number of descriptors' (11). A third arrow points from the 'header' view to a 'Shared Output' view, which displays 'output format' (Y'CbCr 4:2:0), 'width' (848), 'height' (640), 'Histogram metadata' (with a disclosure triangle), 'lux.value' (9.8125), and 'lux.scale' (log2). A final arrow points from the 'Shared Output' view to a 'Histogram metadata' view, which shows 'bHistEnabled' (off), 'bHistSelSrcRGBY' (RGBY), 'sizeHistSelBin' (256), and 'data' (represented by three dots). A large bracket on the left side of the diagram is labeled 'Hierarchical Data Browser', indicating that the entire process of displaying metadata in a hierarchical format is managed by this component.

Hierarchical Data Browser

Jcam includes a custom UITableView that displays arbitrary hierarchical formatted data. A metadata formatter encodes the various field names and values into strings which are displayed in this view.

The data can be formatted as:

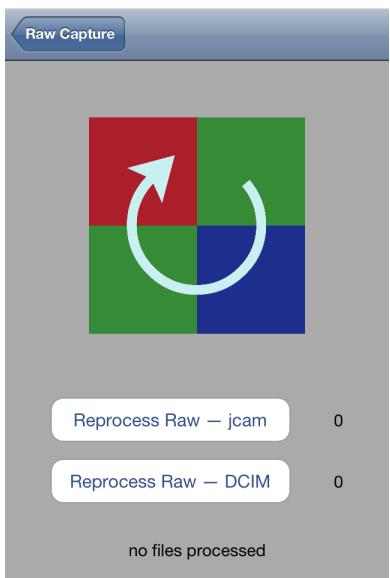
- a signed or unsigned integer
- a hex representation of the integer
- a float (e.g. from a fixed-point value)
- a boolean which can be displayed as either on/off or yes/no
- a string from a specialized formatter, for example, to show a label for enum values
- or, a list-of-values, which forms a new hierarchy of formatted data!

There is no limitation for how many levels can be in the hierarchy, so if the data is logically organized as a struct of structs of structs, it can be three levels deep.

The app really knows nothing about the metadata, other than it is a blob of data. A separate "metadata formatter" does all the "magic" to encode field names and values into strings. A tiny wrapper, a custom Cocoa Touch UITableView Controller, displays the data hierarchy in UITableViewCells.

The formatter can be extended or modified as the metadata changes, without requiring code changes to the rest of jcam, or UI changes in Interface Builder.

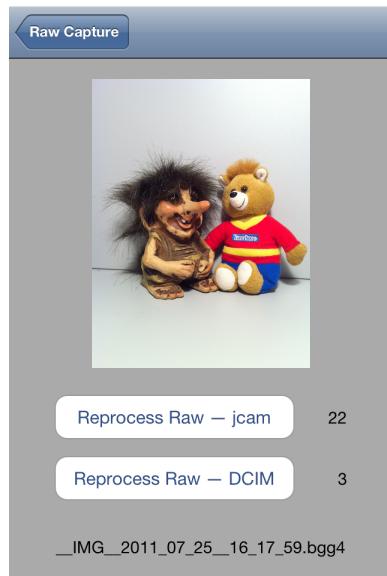
Reprocessing Raw Images (H4 only)



Jcam normally saves the metadata and full-size raw files, but JPEGs may not be full size because the Y'CbCr data is rendered at a lower resolution for realtime previewing to the display. Jcam has the ability to reprocess the saved raw and metadata files to reconstruct full size JPEGs and full-size uncompressed “yuv” files (saved in NV12 format). This is available using the “Reprocess” button in the upper right of the Raw Capture (main) view.

The raw reprocessing will reprocess all raw images saved by jcam at the touch of a button. It can also reprocess raw images

saved by CoreMedia camera apps (stored in /var/mobile/Media/DCIM, when enabled). Each raw file will be saved with the same name with an additional .jpeg extension. For example, the file /some/path/foo.raw, is reprocessed to /some/path/foo.raw.jpeg. As each raw image is reprocessed, it is displayed on the screen, and a counter indicates how many images have been reprocessed (all raw images saved by jcam or CoreMedia are reprocessed each time the corresponding button is pressed).



Tip:

To *enable* CoreMedia raw captures, log in as **mobile** and enter:

```
defaults write com.apple.coremedia enable_raw_photos 1; killall mediaserverd
```

To *disable* CoreMedia raw captures, log in as **mobile** and enter:

```
defaults delete com.apple.coremedia enable_raw_photos; killall mediaserverd
```

Change History

Version 3.0.0

- Submitted to [Peace16A28x](#)
- <rdar://35235280> Modernize jcam for Emet+ (remove legacy cruft)
- <rdar://32184593> CrashTracer: jcam at DYLD error: Library not loaded: H1oISPServices referenced from jcam expected in None with reason image not found
- <rdar://40195696> jcam: Workaround UIKit button title update regression, [rdar://14130081](#)

Version 2.9.1

- Submitted to [Tigris15A253](#)
- <rdar://30320022> Jcam: Request to support processing raw with *.rgg4 file name
- Added support to reprocess raws from back-telephoto (in /var/mobile/Media/DCIM/)

Version 2.9.0

- Submitted to [Tigris15A192](#)
- <rdar://29528950> jCam/Tigris: remove dependency on H4ISPServices.framework
- Added support for stream control APIs: FigCaptureDeviceRequestControlOfStreams & FigCaptureDeviceRelinquishControlOfStreams.

Version 2.8.9

- Submitted to [Erie14E168](#)
- <rdar://28016318> JCAM: a new entitlement will be required to access Camera Driver IOKit API
- Added support for H1oISP.mediacapture in LoadAndCreateVideoCaptureDevice.

Version 2.8.8

- Submitted to [Whitetail14A306](#)
- Changed the display pipeline to no longer rely on CALayer APIs which used global IOSurfaceIDs, <rdar://27186032>. This was necessitated by <rdar://27107243> in Whitetail14A304, which broke the CAImageQueue display code.

Version 2.8.7

- Submitted to [Whitetail14A180](#)
- <rdar://24123854> jcam: on H9 (Cayman) products, load the H9ISP.mediacapture plug-in instead of H6ISP.mediacapture plug-in
- <rdar://11779063> jcam: Metadata is "not available" when tap on thumbnail
- Filtered streams array to use front and back camera (e.g. for D11).
- Worked around suspend/resume changes in Whitetail.
- Does anyone ever read this? If you are the first person to e-mail jay541 "at" apple "dot" com before February 29, 2016, with the subject "jcam free meal", you can claim a free \$10 Caffè Macs meal ticket¹.

Version 2.8.6

- Submitted to [Monarch13A303](#)
- Fixed <rdar://21773480> jcam: BracketedCaptureViewController is broken under Monarch

Version 2.8.5:

- Submitted to [Copper12H28/Monarch13A172](#)
- Removed AspenSDK.xcconfig: <rdar://problem/19317385> jcam: Remove usage of deprecated configs and build settings: AspenSDK.xcconfig
- Support for 256 and 1024 frame averaging using 32-bit sample accumulation buffers: <rdar://problem/16295439> jcam: average more than 64 frames (user specified number) for low noise.
- Fixed metadata browser: <rdar://problem/19693895> jcam: Metadata view formatting is munged by Okemo UITableView change

Version 2.8.4:

- Submitted to [OkemoTaos12B381/Blacktail12E100](#)
- Worked around regression in AppleH6CameraInterface: <rdar://17984685> JCam: On H7 devices, capture does not finish if focus is set to "AF at capture".

¹ You are not eligible if you claimed a free meal ticket in the prior 12 months.

- Worked around apparent UILabel-related regression in Okemo/OkemoTaos, <[rdar://18185591](#)>.
- Added iOS 8 Launch Image XIB file for N56/N61.

Version 2.8.3:

- Submitted to Okemo12A330/Blacktail12C62
- Added Focus Sweep limits feature enhancement, <[rdar://problem/16295041](#)>
- Added support for N56 & N61 (adopt Auto Layout), <[rdar://problem/17075202](#)>
- Added compatibility with new CoreMedia camera release behavior, <[rdar://problem/17008983](#)>
- Stopped jcam from automatically choosing “experimental” camera formats, <[rdar://problem/17082341](#)>

Version 2.8.2:

- Submitted to Sochi11D109/Okemo12A150
- Fixed Focus Sweep settings view .xib file, <[rdar://problem/15036262](#)>
- Worked around an apparent iOS 7 bug in which a QLPreviewController clobbered the translucency state of the navigationBar, which in turn hid portions of other views, <[rdar://problem/15540471](#)>.

Version 2.8.1:

- Submitted to Sochi11D68
- Worked around autofocus & bracketed capture race condition, <[rdar://problem/15036262](#)>
- Reverted .xib files to Xcode 4.6, to be compatible with older B&I tools, as per Rob M.
- Fixed .xib file broken by Xcode 4.6 → 5.0 → 4.6 conversion

Version 2.8:

- Submitted to Sochi11D60
- Worked around iOS UIKit regression, <[rdar://problem/13986296](#)>
- Fixed up .xib files, that was munged by recent InterfaceBuilders.
- New Focus View in response to <[rdar://problem/14503420](#)>

Version 2.7:

- Submitted to Innsbruck11A398
- Added flicker detect status, <[rdar://problem/13985284](#)>
- Added ALS T-parameter monitor, <[rdar://problem/13985307](#)>
- Changed Color Histogram to a multi-purpose status monitor. Tapping it will transition from color histogram, to flicker detect status (60Hz is blue, 50Hz is magenta), to an ALS T-parameter monitor. Then when tapped again it will return to displaying the Color Histogram.
- Worked around UIKit regressions, <[rdar://problem/13665633&13665634](#)>

Version 2.6.3:

- Submitted to Innsbruck11A338
- Enabled adaptive tone curve on H6, <[rdar://problem/12039250](#)>
- Fixed <[rdar://problem/13240532](#)> N78A/Innsbruck11A283; jCam crashes at launch

Version 2.6.2:

- Submitted to Innsbruck11A334
- Workaround for intermittent reprocessing bug, <[rdar://problem/13617348](#)>

Version 2.6.1:

- Submitted to Innsbruck11A327
- Fixed unlock AE bug, <[rdar://problem/13572221](#)>

Version 2.6:

- Submitted to Innsbruck11A320
- Added support for H6
- Addressed a number of H6/Innsbruck related radars: <[rdar://problem/13341842&13188534&13144132&13483572&12229917&12163150&12239984](#)>

Version 2.5.1:

- Submitted to Sundance10A373
- Worked around DART issue, <[rdar://problem/11537695&11969732](#)>

Version 2.5:

- Submitted to Sundance10A366
- Added “focus sweep” mode, <[rdar://problem/1178194&11782152](#)>
- Fixed issue where JPEGs were not always saved, <[rdar://problem/11782159](#)>
- Added option to enable histograms in metadata, <[rdar://problem/11872414](#)>
- Changed launch screen to prevent buttons from showing up before app was ready to process events.

Version 2.4:

- Submitted to Sundance10A336
- Add UI in “Settings” to select “focus” button behavior: auto/manual, <[rdar://problem/11671674](#)>

Version 2.3:

- Submitted to Sundance10A324
- Added “fast raw capture” mode with packed raw files, <[rdar://problem/11476796](#)>
- Added activity indicator to reprocess raw view
- Fixed C++11 conformance issues, <[rdar://problem/11526992](#)>
- Disable the iOS sleep timer during repeated captures and while reprocessing to allow captures and reprocessing to complete, <[rdar://problem/11531720](#)>

Version 2.2:

- Submitted to Sundance10A307
- Added double tap of camera view to zoom in and out of full-screen mode, <[rdar://problem/10791377](#)>

Version 2.1.1:

- Submitted to Sundance10A303
- Fixed reliability issue with repeated captures, <[rdar://problem/10696752](#)>
- Fixed bracketed capture issue in which the actual bracketed capture sequence did not always match the bracketed capture sequence chosen from the menu, <[rdar://problem/11389898](#)>
- Improved documentation based on feedback from users.

Version 2.1:

- Submitted to Hoodoo9B129, Sundance10A163
- Exposed multi-frame repeat capability in UI
- Added user selectable delay between repeated captures
- Added JPEG-only capture mode
- Enhanced feedback on capture button
- Worked around <[rdar://problem/9882333](#)>

Version 2.0:

- Submitted to Hoodoo9B97, Sundance10A131
- DRC — now enabled by default (currently there is no UI switch for this).
- AE Algorithm — FD assisted AE is now enabled using FW defaults for AEFaceDetectionMetering parameters. There is also a UI switch to turn this on/off.
- exposure limits — MinimumFrameRate was arbitrarily set to 5. It is now set to 15. This should now match the camera app.
- various pre-configured bracketed capture modes are available
- a user-configurable bracketed capture mode is available
- flash capture mode is available
- longer capture sequences (128 & 256 frames) are available
- improved UI with a “capture mode...” button on the main view, instead of in settings
- Color Histogram auto updates are disabled during capture in some modes, to avoid/minimize frame drops.
- Fixed flash capture bug from earlier beta, <[rdar://problem/10113528](#)>
- “AWB Scheme” select was removed from the UI
- Raw reprocessing will also save out uncompressed “yuv” in NV12 format
- Minor bug fixes and robustness improvements

Version 1.6 (not submitted to any build):

- Added 128 and 256 frame durations to multiframe capture menu
- Tweaked some streaming parameters to more closely match camera.app
- Made the Reprocess view scrollable

Version 1.5:

- Submitted to Telluride9A298, Hoodoo9B51, Sundance10A85
- Added EXIF data to saved JPEGs.

Version 1.4:

- Submitted to Telluride9A290, Hoodoo9B45, Sundance10A79
- The powerblur algorithm is applied to reprocessed images
- Raw images with DRC enabled are reprocessed twice, and the final image is saved (with powerblur)
- Added optional test pattern overlay

Version 1.3:

- Submitted to Telluride9A284, Hoodoo9B41, Sundance10A75
- Added raw image reprocessing

Version 1.2:

- Submitted to Telluride9A263, Hoodoo9B26, Sundance10A61
- Added multiframe capture and frame averaging to Settings panel
- AE freeze during averaged captures
- Default AWB scheme now set to AWBi
- Provided better feedback in capture button (capture/average and number of frames)
- Fixed some issues from llvm static analyzer
- Mach-banding removed from Color Histogram color table
- Updated User Guide

Version 1.1:

- Submitted to Telluride9A237, Hoodoo9B4, Sundance10A37
- Changed metadata formatter to use `H4ISPServices`, so it should always display the current metadata for Telluride9A225 and later.
- The User Guide is now available from Settings within the app.

Version 1.0:

- First “feature complete” release, not submitted to any build
- ability for the user to select the stream format for the front and back cameras.
- an interactive hierarchical metadata browser.
- the capture button also reflects the selected AWB scheme.
- an initial linear ramp was added as a “legend” for the color histogram.