Typical Production Video Camera

Partial OCA object model

This is a casual example. No claims are made as to correctness or appropriateness of design. It's mainly to demonstrate how a video device's parameters can be rendered into an object model.

Jeff Berryman
ja.berryman@us.bosch.com
+1 952 457 5445

version 5

PRODUCTION NOTES - Jeff

What kind of camera is this?

The camera is a current HD/4k model from a well-known manufacturer.

How did I make the OCA model?

I made the OCA model from information in the control menu descriptions in the product's user manual.

I went through two of the six menus and guessed at what parameters users would want to be network-controllable. For example, a I'd make a parameter like "video gain" network-controllable, but something like "viewfinder brightness" I assumed would just be an operator preference that didn't require remote control.

I think about 75% of the menu options ended up as network-controllable.

I didn't go through all the menus - I just picked two of them that seemed representative and interesting.

I didn't have any information about other camera parameters not in the menus, so if there are internal operating controls that should be network-controllable, they won't be in this model.

How well did the current OCA fit the model?

As we know, the current OCA has no video-specific control classes or datatypes. In a few cases, I needed to invent new classes and datatypes to make appropriate interfaces, but I was surprised at how seldom this was necessary.

The new classes and datatypes are described below, at the end of the model.

When we do extend OCA to include video classes, there will probably be more video-specific classes and datatypes that could be used to make this model more elegant and concise.

TYPICAL PRODUCTION VIDEO CAMERA - Partial OCA Object Model

Basic Device Layout - Managers and Root Block

DEVICE

OcaDeviceManager	DeviceManager
OcaSubscriptionManager	SubscriptionManager
OcaMediaClockManager	MediaClockManager
OcaSecurityManager	SecurityManager

Root Block

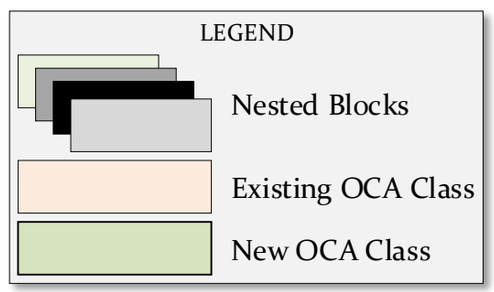
All functional control and monitoring elements are inside the Root Block.

See the following pages for details.

For simplicity, this example excludes video and audio connection management features.

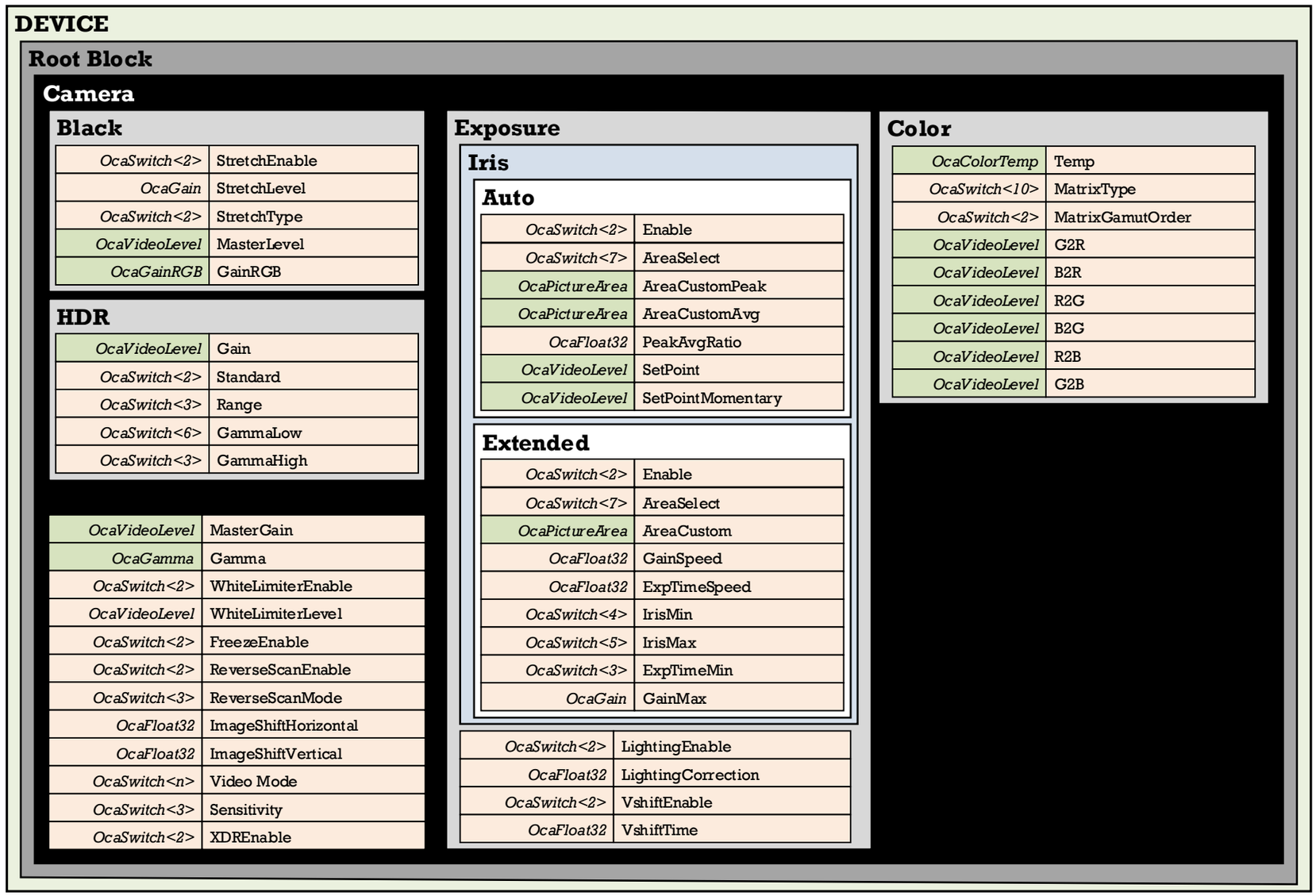
In practice, connection management might be implemented by mechanisms defined in:

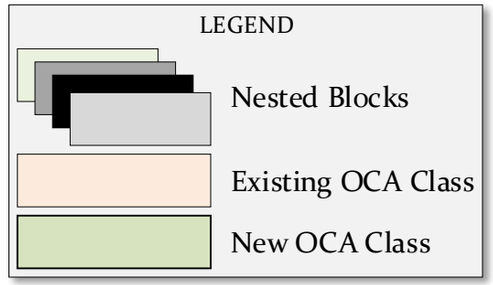
- *The NMOS Suite;*
- *OCA-CM3, the OCA connection management feature set; or*
- *NCA, the harmonized NMOS-OCA scheme currently being developed by the OCA Alliance.*



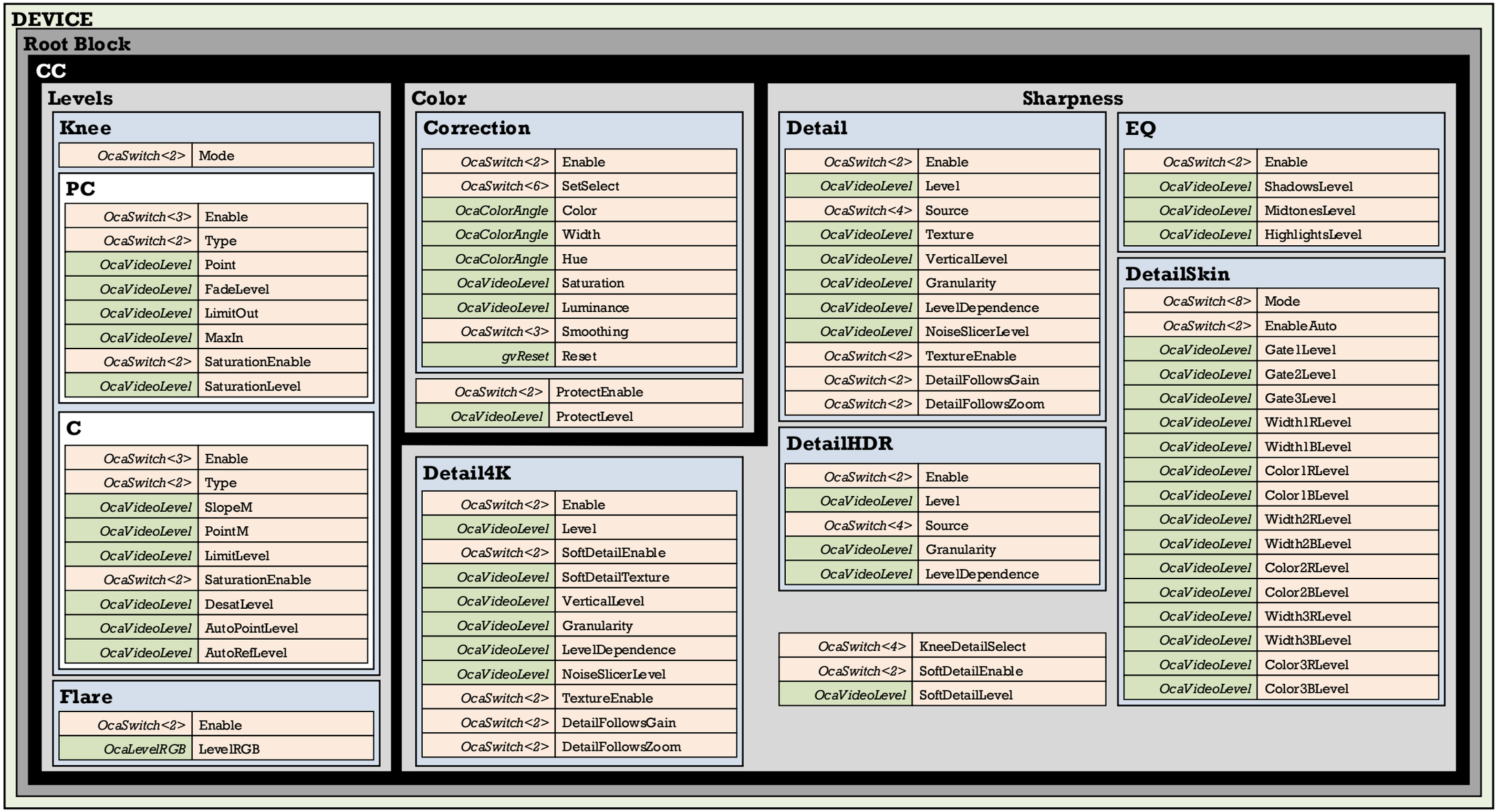
TYPICAL PRODUCTION VIDEO CAMERA - Partial OCA Object Model

Production Control Feature Set





TYPICAL PRODUCTION VIDEO CAMERA - Partial OCA Object Model
Creative Control Feature Set



TYPICAL PRODUCTION VIDEO CAMERA - Partial OCA Object Model

New classes and datatypes needed that are not part of current OCA object model

New Classes

OcaGamma *- inherits from* ***OcaWorker***	*OcaGammaCurve* .Curve *OcaGammaPreset* .Preset *OcaVideoLevel* .Level *OcaLevelRGB* .LevelRGB *OcaStatus* Get|SetCurve(...) *OcaStatus Get|SetPreset(...)* *OcaStatus Get|SetLevel(...)* *OcaStatus Get|SetLevelRGB(...)*
gvReset *- inherits from* ***OcaAgent***	*OcaStatus* Reset(...)

New Datatypes

OcaVideoLevel	*OcaFloat32* //min=0, max=100
OcaRGBLevel	*OcaFloat32* .R *OcaFloat32* .G *OcaFloat32* .B
OcaGammaCurve	*enum {BBC04, BBC05, BBC06, ARD, ITU709, Gamma-J, Gamma-S}*
OcaGammaPreset	*enum {nom, lin, var}*
OcaColorTemp	*OcaFloat32* .Filter *OcaFloat32* .Temp *OcaFloat32* .Tint
OcaPictureArea	.*OcaFloat32* .Top .*OcaFloat32* .Bottom .*OcaFloat32* .Left .*OcaFloat32* .Right