

FAQ 1: If I embrace a common control model, will I have to give away all my unique selling points ("secret sauce")?

1. **No.** A device control model defines control APIs. A device's network control API is simply a virtual control panel. It doesn't reveal any more secret sauce than a physical control panel would.
2. **No.** A properly designed device control scheme won't require devices to reveal *all* of their control points in their public network control API. You can hide things.

FAQ 2: If I have an object-oriented control model, will I have to rework my device's software/firmware to be object-oriented?

- ◆ **No, you won't.** The object-oriented device control model only describes the device's network control API. What's under the hood can be anything, as long as the network API is aligned with the control model.

FAQ 3: Products are many and varied. How can a common control model be feasible?

- ◆ **The trick is not to define too much.** Defining huge canonical standard models for cameras, switchers, mixers, matrices, and so on is infeasible. However, standardizing components and subassemblies is feasible. The electronics industry does it all the time.

FAQ 4: I already have a proprietary control scheme. How can I afford to rework all my products to support a new standard one?

- ◆ **You can't.** Instead, you should define a software interface (aka gateway aka proxy aka translator) from your protocol to the standard one, and deploy it at points in your product line where you want to interoperate with third-party equipment.

Going forward, you might consider whether new product lines will stick with your proprietary scheme or take the leap to the new one. That's a case-by-case business decision.

FAQ 5: OMG, the control object model specification is huge. How will it run in my microcontrollers? Also, how will my programmers learn it quickly?

1. Device control object models are large because there are many different kinds of devices doing many different kinds of things. However, ***you only have to implement what you use***. For example, the AES70 model has about 75 different classes, but you can run AES70 in an Arduino.
2. Properly designed control object models may have many elements, but those elements will be based on a small and clean framework. Your programmers need to learn the framework. They can retrieve the details as they require them.

FAQ 6: I like object model [x] but the associated control protocol doesn't work for me.

example 1: I have a web-based application, but the control protocol associated with the model isn't web-oriented.

example 2: I have a non-web-based application, but the control protocol includes web overheads I don't need.

1. The important thing for interoperability is to have a common device control object model. **However, not everyone has to use the same protocol for accessing that model.** Translation from one protocol to another is straightforward when both protocols are working from the same model.
2. Protocol design is to some extent a religious issue. When you're evaluating an object model, it's good to keep protocol discussions aside most of the time. If needed, they should come later.

FAQ 7: I have a big production mission, with workflow management requirements, complex media flows, intellectual property control requirements, asset scheduling and management needs, multiple site coordination requirements, artist management issues, lots of security concerns, etcetera etcetera. How in the world can a common device control model handle all of that?

- ◆ **It can't! But it can support functions that do.** Device control is *only* control (and monitoring) of devices. Other business functions are built on top of it, in a higher functional layer that has been called the Production Management layer, the Orchestration layer, the Workflow layer, or other things.
- ◆ In the future, many of these requirements will be fulfilled by microservices running in this higher layer. In order to be effective, these higher-layer services will need a uniform way of controlling the devices they use - this is one big reason for wanting a common device control model.

- Jeff Berryman
ja.berryman@us.bosch.com
May 1, 2020