

AES70

Jeff Berryman
AES Standards vice-chair
2021.11.15

What is AES70?

What's in AES70?

- An object-oriented framework for control interfaces that audio devices present to a data network;
- A standardized device object model for controllable devices;
- A rich and extensible repertoire of control class definitions (the **AES70 Object Model**) that represent the signal processing, control logic, and network connection functions of modern audio devices
- An application protocol called OCP.1 ("Open Control Protocol 1") that defines command and response formats and sequences for control and monitoring of OCA-compliant devices over IP networks.

What's *not* in AES70?

- Audio program transport;
- A programming model for OCA-compliant devices;
- A user interface definition or generation scheme for OCA-compliant devices;
- Standardized semantics for controllable elements (e.g. standard filter shapes);
- Standard device profiles (e.g. "standard mixer", "standard power amp").

AES70 defines the set of APIs a device exposes to the network.

The AES70 Family

Core

- AES70-1 **AES70 Framework.** Text document that defines the basic AES70 mechanisms for control and monitoring.
- AES70-2 **AES70 class structure.** Text + UML document that defines AES70's control & monitoring repertoire. UML stands for Universal Modeling Language.
- AES70-3 **AES70 binary protocol.** Text + UML document that defines OCP.1, a binary protocol for using AES70 over IP networks.
- AES70-4 **AES70 JSON. NEW.** Text + UML document that defines OCP.2, a JSON protocol for using AES70 over IP networks.

Adaptations

- AES70-21 **AES67 Adaptation. NEW.** This standard will specify the use of AES70 connection management for AES67 stream transport connections.
- AES70-22 **MILAN Adaptation. NEW.** This standard will specify the use of AES70 connection management for MILAN stream transport connections.

Versions

Core

AES70-1	AES70 Framework	Current: AES70-1-2018	Mar 2022: AES70-1-2022
AES70-2	AES70 class structure	Current: AES70-2-2018	Mar 2022: AES70-2-2022
AES70-3	AES70 binary protocol	Current: AES70-3-2018	Mar 2022: AES70-3-2022
AES70-4	AES70 JSON	Current: none	Q2 2022: AES70-4-2022

Adaptations

AES70-21	AES67 Adaptation	Current: none	Q2 2022: AES70-21-2022.
AES70-22	MILAN Adaptation	Current: none	Q2 2022: AES70-22-2022

- What's "OCA"?
 - OCA stands **for Open Control Architecture**
 - the technology that AES70 standardizes.
 - originally developed by the OCA Alliance trade association
 - now standardized by the AES70 family
 - OCA still exists and has become an ecosystem that surrounds AES70
 - maintained by the OCA Alliance (<https://ocaalliance.com>)
 - technical site at <https://ocaalliance.github.io/>
 - includes presentations, tutorials, white papers, open-source code, and more

What's good about AES70?

AES70 is the only control architecture standard that is all of the following:

- Open and license-free
- Pro application oriented
- Scalable up to huge network sizes
- Suitable for mission-critical applications
- Friendly to proprietary product features
- Futureproof
- Secure
- Able to support dynamic DSP device reconfiguration
- Heterogeneous-network capable

What's in AES70?

Objects

Categories of Control Objects

- **Managers** Standard housekeeping objects, mostly the same in every device.
- **Workers** Objects that correspond to audio processing control functions.
- **Agents** Devices that provide various control functions or modify the control command stream, but do not map directly to signal processing elements.
- **Networks** AES70's connection management feature set.

Elements of Control Objects

- **Properties** Variables that define the state of the object
- **Methods** Operators that change properties and cause actions
- **Events** Signals emitted by objects to indicate state changes
- **Object Number** (aka ONo) Unique identifier of object within the device

AES70 control objects are abstractions that define a device's network API. They may or may not correspond one-for-one with software or hardware elements of the device. For example, a master gain control object may in fact control several real gain-setting elements in the device.

Classes

- Templates from which control objects are created.
- Every class is uniquely identified by a **class ID**, a structured identifier used in various ways throughout AES70.
- AES70 classes inherit elements in the standard object-oriented manner. Only simple inheritance is supported.
- The set of OCA classes (aka "AES70 Object Model aka "AES70 Class Tree" aka "OCC") defines AES70 's functional repertoire.
- The object model will evolve over time to accommodate new device types and new manufacturers.
- Object model inheritance rules create a constrained evolution regime that maximizes upward compatibility and ensures graceful evolution through orderly class specialization.
- Object model inheritance rules support the addition of proprietary classes to the class tree in a way that maximizes compatibility with the standard classes

Events

- Event: transient state of an object that can cause it to send one or more event notification messages.
- Events have class-specific types. Each class may have a repertoire of events of various types.
- Event definitions are inherited.
- The most commonly used event is [OcaPropertyChanged](#), an event that causes an object to emit a notification whenever a value of any of its properties changes.
 - [OcaPropertyChanged](#) is an event of the root class [OcaRoot](#), and is therefore defined for all classes in the tree.
- Notifications are sent only to subscribing objects.
- Subscriptions are registered with and managed by the Subscription Manager.

AES70 Object Model

AES70's Control & Monitoring Repertoire

Object Model Overview

Workers

Classes that deal with audio processing

Actuators *Classes that control audio processing*

Sensors
the device

Classes that monitor

Blocks and Matrices

Classes that define device control and processing groups

Agents *Classes that affect the flow and timing of control*

Networks *Connection management classes*

Managers

Device housekeeping classes

Workers

OcaActuator	<i>Base class for classes that control audio processing</i>
OcaMute	<i>Signal mute</i>
OcaPolarity	<i>Signal inversion</i>
OcaSwitch	<i>1 of n selector</i>
OcaGain	<i>Simple gain in dB</i>
OcaPanBalance	<i>Pan or balance control</i>
OcaDelay	<i>Signal delay in mSec</i>
OcaDelayExtended	<i>Signal delay in mSec, ft, m</i>
OcaFrequencyActuator	<i>Frequency</i>
OcaFilterClassical	<i>Bessel, Butterworth, etc.</i>
OcaFilterParametric	<i>Peaking or shelving parametric filter</i>
OcaFilterPolynomial	<i>Rational polynomial filter</i>
OcaFilterFIR	<i>FIR specified by coefficients</i>
OcaFilterArbitraryCurve	<i>Magnitude vs freq curve</i>
OcaDynamics	<i>Generalized compressor/expander</i>
OcaDynamicsDetector	<i>Side-chain detector</i>
OcaDynamicsCurve	<i>Dynamics input vs output level curve</i>
OcaSignalGenerator	<i>Multi-waveform signal generator</i>
OcaSignalInput	<i>Device signal input port</i>
OcaSignalOutput	<i>Device signal output port</i>
OcaTemperatureActuator	<i>Temperature parameter</i>
OcaIdentificationActuator	<i>Device identification light or other flag</i>

Actuators, continued

OcaBasicActuator	<i>Base class for weakly typed actuators</i>
OcaBooleanActuator	<i>Weakly typed actuators ...</i>
OcaInt8Actuator	...
OcaInt16Actuator	...
OcaInt32Actuator	...
OcaInt64Actuator	...
OcaUInt8Actuator	...
OcaUInt16Actuator	...
OcaUInt32Actuator	...
OcaUInt64Actuator	...
OcaFloat32Actuator	...
OcaFloat64Actuator	...
OcaStringActuator	...
OcaBitStringActuator	...

Workers

OcaSensor	<i>Base class for classes that monitor the device</i>
OcaLevelSensor	<i>Signal level</i>
OcaAudioLevelSensor	<i>Audio level with standard meter laws</i>
OcaTimeIntervalSensor	<i>Time interval</i>
OcaFrequencySensor	<i>Frequency</i>
OcaTemperatureSensor	<i>Temperature</i>
OcaIdentificationSensor	<i>Monitors a button push or something</i>
OcaBasicSensor	<i>Base class for weakly typed sensors for general use</i>
OcaBooleanSensor ...	
OcaInt8Sensor ...	
OcaInt16Sensor ...	
OcaInt32Sensor ...	
OcaInt64Sensor ...	
OcaUInt8Sensor ...	
OcaUInt16Sensor ...	
OcaUInt32Sensor ...	
OcaUInt64Sensor ...	
OcaFloat32Sensor ...	
OcaFloat64Sensor ...	
OcaStringSensor ...	
OcaBitStringSensor ...	

Workers

Blocks and Matrices

Classes that allow grouping of device functions

OcaBlock *Container for Workers, Agents, and Networks that defines a related set of device functions*

OcaBlockFactory *Constructor for OcaBlock objects; to be used with dynamically-reconfigurable DSP devices*

OcaMatrix *Specialized container for 2-dimensional arrays of processing elements; superset of conventional gain matrix.*

Networks *Connection management classes*

OcaApplicationNetwork *Abstract base class for other network classes*

OcaControlNetwork *Application network for transport of control traffic (e.g. an AES70 network)*

OcaMediaTransportNetwork *Application network for transport of media content (e.g. an AES67 network)*

AES70 Device Model

The AES70 Device Model is a deployment of objects defined by classes in the AES70 Object Model. It defines the basic object configuration of all AES70-compliant devices. In practice, other objects will be instantiated from the Object Model to represent the specific functions of each device.

Device Model

REQUIRED MANAGERS

Device Manager

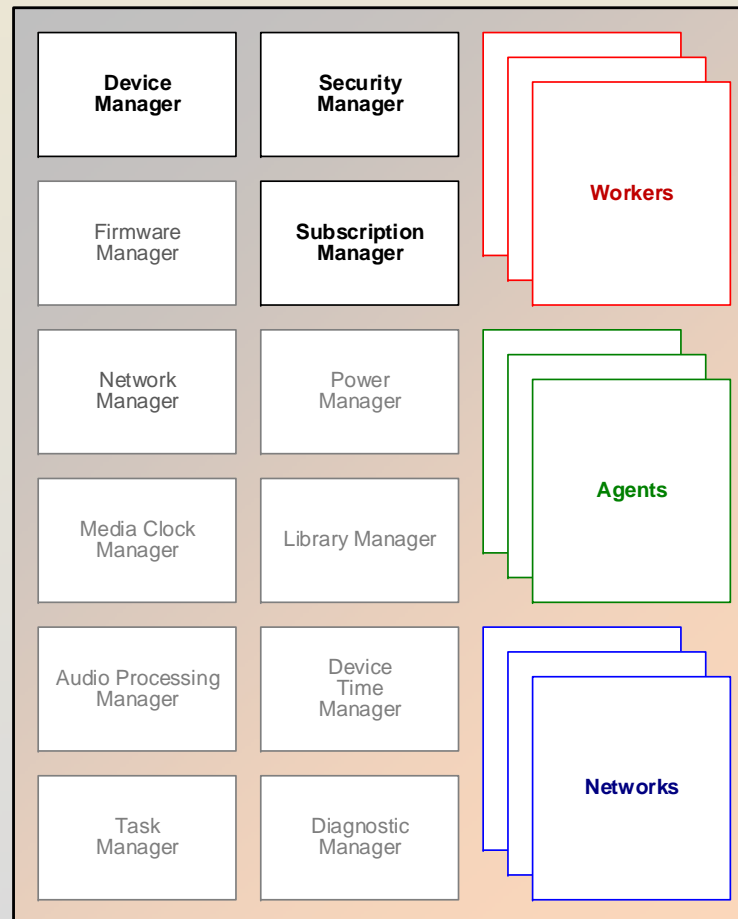
Manages information relevant to the whole device.

Security Manager

Manages security keys.

Subscription Manager

Manages event subscriptions.



OPTIONAL MANAGERS

Power Manager

Manages power supplies and batteries.

Firmware Manager

Manages firmware versions and, optionally, updates.

Network Manager

Manages connection(s) to network(s).

Media Clock Manager

Manages media clocks.

Library Manager

Manages stored parameter settings.

Audio Processing Manager

Holds global signal processing parameters.

Power Manager

Manages power supplies and batteries.

Device Time Manager

Manages time reference objects.

Task Manager

Manages stored processing sequences.

Diagnostic Manager

Offers features to help installation and setup.

Sneak peek at AES70-2022

- **Media files**
Controllers can manage device audio file recording, playback, and storage.
- **Logging**
Devices can generate logs of all kinds that controllers can retrieve later.
- **Command sets**
Devices can store predefined control command sequences that can be executed later - on controller request, at scheduled times, or at predefined points in designated media streams.
- **Streamlined media connection management**
Managing device media streaming connections is now more powerful and easier to program.
- **Connection negotiation support**
There's flexible new support for *connection negotiation*, the process by which controllers develop compatible transmitting and receiving settings for media stream connections.
- **New support for handling stored parameters**
AES70-2018's mechanism for storing operating parameter settings ("presets") inside devices and recalling them later has been replaced with a more powerful scheme capable of handling complex devices with tens of thousands of parameters.
- **Dozens of small additions, ease of use improvements, and documentation improvements**

[AES70-2022 will be upwards-compatible from the current AES70-2018.](#)

Thanks for your interest in AES70 !

Jeff Berryman
AES Standards vice-chair
2021.11.15
ja.Berryman@us.bosch.com