Secretariat 2015/11/12 14:49 DRAFT REVISED AES70-2-xxxx

STANDARDS AND INFORMATION DOCUMENTS

Call for comment on DRAFT AES standard for Audio applications of networks - Open Control Architecture - Part 2: Class structure

This document was developed by a writing group of the Audio Engineering Society Standards Committee (AESSC) and has been prepared for comment according to AES policies and procedures. It has been brought to the attention of International Electrotechnical Commission Technical Committee 100. Existing international standards relating to the subject of this document were used and referenced throughout its development.

Address comments by E-mail to standards@aes.org, or by mail to the AESSC Secretariat, Audio Engineering Society, 60 E 42nd St., New York NY 10165. **Only comments so addressed will be considered**. E-mail is preferred. **Comments that suggest changes must include proposed wording.** Comments shall be restricted to this document only. Send comments to other documents separately. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

This document will be approved by the AES after any adverse comment received within **six weeks** of the publication of this call on http://www.aes.org/standards/comments/, **2012-11-12**, has been resolved. Any person receiving this call first through the *JAES* distribution may inform the Secretariat immediately of an intention to comment within a month of this distribution.

Because this document is a draft and is subject to change, no portion of it shall be quoted in any publication without the written permission of the AES, and all published references to it must include a prominent warning that the draft will be changed and must not be used as a standard.

[This page intentionally blank]

DRAFT

AES standard for Audio applications of networks - Open Control Architecture - Part 2: Class structure

Published by
Audio Engineering Society, Inc.

Abstract

AES70 defines a scalable control-protocol architecture for professional media networks. It addresses device control and monitoring only; it does not define standards for streaming media transport. However, OCA is intended to cooperate with various media transport architectures.

AES70 is divided into a number of separate parts. This Part 2 specifies the control class structure for AES70 that defines the AES70 control and monitoring functional capabilities and should be read in conjunction with Part 1, Framework, and Part 3, TCP/IP communications protocol.

Table of Contents

Foreword

This foreword is not part of this document, AES70-2-xxxx, *AES standard for Audio applications of networks - Open control architecture - Class Structure*.

This document is a member of the three-document set that defines AES70, the Open Control Architecture (OCA). AES70-2 defines the control class structure for AES70. Other parts define the overall framework and the specific protocols used.

The development project for this standard was originally proposed by the Open Control Architecture Alliance (OCA Alliance) and initiated in October 2012 as project AES-X210 to be developed in task group SC-02-12-L. The OCA Alliance also contributed the task-group working draft and, as a direct result, there are a number of references to "OCA" in the protocol in order to maintain compatibility with implementations already in the field.

The members of the writing group that developed this document in draft are: J. Berryman, H. Hamamatsu, T. Head, S. Jones, M. Lave, N. O'Neill, M. Renz, M. Smaak, G. van Beuningen, S. van Tienen, E. Wetzell.

J. Berryman led the task group.

Richard Foss
Chair, working group SC-02-12
2015-11-12

Note on normative language

In AES standards documents, sentences containing the word "shall" are requirements for compliance with the document. Sentences containing the verb "should" are strong suggestions (recommendations). Sentences giving permission use the verb "may". Sentences expressing a possibility use the verb "can".

DRAFT

AES standard for Audio applications of networks - Open control architecture - Class Structure

0 Introduction

0.1 General

This document defines the class structure of the Open Control Architecture (OCA) for the control and monitoring of media networks. The class structure defines the control repertoire. In what follows, the class structure is referred to as the AES70 *OCC*.

The elements of the AES70 OCC are class definitions in the object-oriented design sense. Each class defines a particular control or monitoring interface element that is accessible over the media network via one or more communications protocols that AES70 defines. An AES70-controllable device may implement a set of these interface elements; the complete set constitutes the interface the device presents to the network for remote control and monitoring purposes. This interface is called the AES70 *device model* and is defined in AES70-1.

To distinguish OCC classes from programming classes, this standard may where appropriate refer to OCC classes as *control classes*, and their instances as *control objects*, where it should be understood that "control" includes both control and monitoring functions.

AES70 specifies system control and monitoring only. It may be integrated with any streaming media transport scheme, as long as the underlying communication network is capable of carrying AES70 control and monitoring traffic.

AES70 does not define a complete device implementation model. For example, if a particular implementation element has no remotely controllable features, then that element is not represented in the AES70 device model.

0.2 Documentation conventions

Numerical values are decimal unless otherwise stated.

A Courier typeface is used to identify **`class names`** to distinguish them from regular text.

Where new terminology is first introduced in body text, the term will be set in an italic typeface.

1 Scope

AES70 defines a scalable control-protocol architecture for professional media networks. It addresses device control and monitoring only; it does not define standards for streaming media transport.

AES70 is divided into a number of separate parts. This Part 2 specifies the control class structure for AES70 that defines the control and monitoring functional capabilities of the standard and should be read in conjunction with Part 1, Framework.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

AES70-1 *AES standard for audio applications of networks - Open Control Architecture - Framework*, Audio Engineering Society, New York, NY., US.

AES70-3 *AES standard for audio applications of Networks - Open Control Architecture - Part 3: Protocol for TCP/IP Networks*, Audio Engineering Society, New York, NY., US.

ISO/IEC 19503 Ed.1:2005 *Information technology – XML Metadata Interchange (XMI)*, International Organization for Standardization (ISO), Geneva, Switzerland.

3 Terms, definitions and abbreviations

For the purposes of this document, the following terms, definitions, and abbreviations apply.

See AES70-1, clause 3.

4 Class structure

The class structure shall be defined by a Universal Modeling Language (UML) document in XML Metadata Interchange (XMI) 2.1 format as defined in ISO/IEC 19503.

See annex A for access data.

NOTE 1 This XMI machine-readable format is intended to enable implementers to have direct access to the class model with maximum speed and the minimum risk of transcription errors compared with building individual class models from a traditional paper description.

NOTE 2 The XMI class model can be parsed in a suitable UML application, such as Enterprise Architect from Sparx Systems.

5 Informative overview

5.1 General

This section gives a brief overview of the AES70-2 OCC.

5.2 Control classes

AES70-2 defines three categories of control classes, as follows:

Workers	Classes that represent signal processing and monitoring functions.
Agents	Classes that represent control-flow processing functions.
Managers	Classes that represent device housekeeping functions.

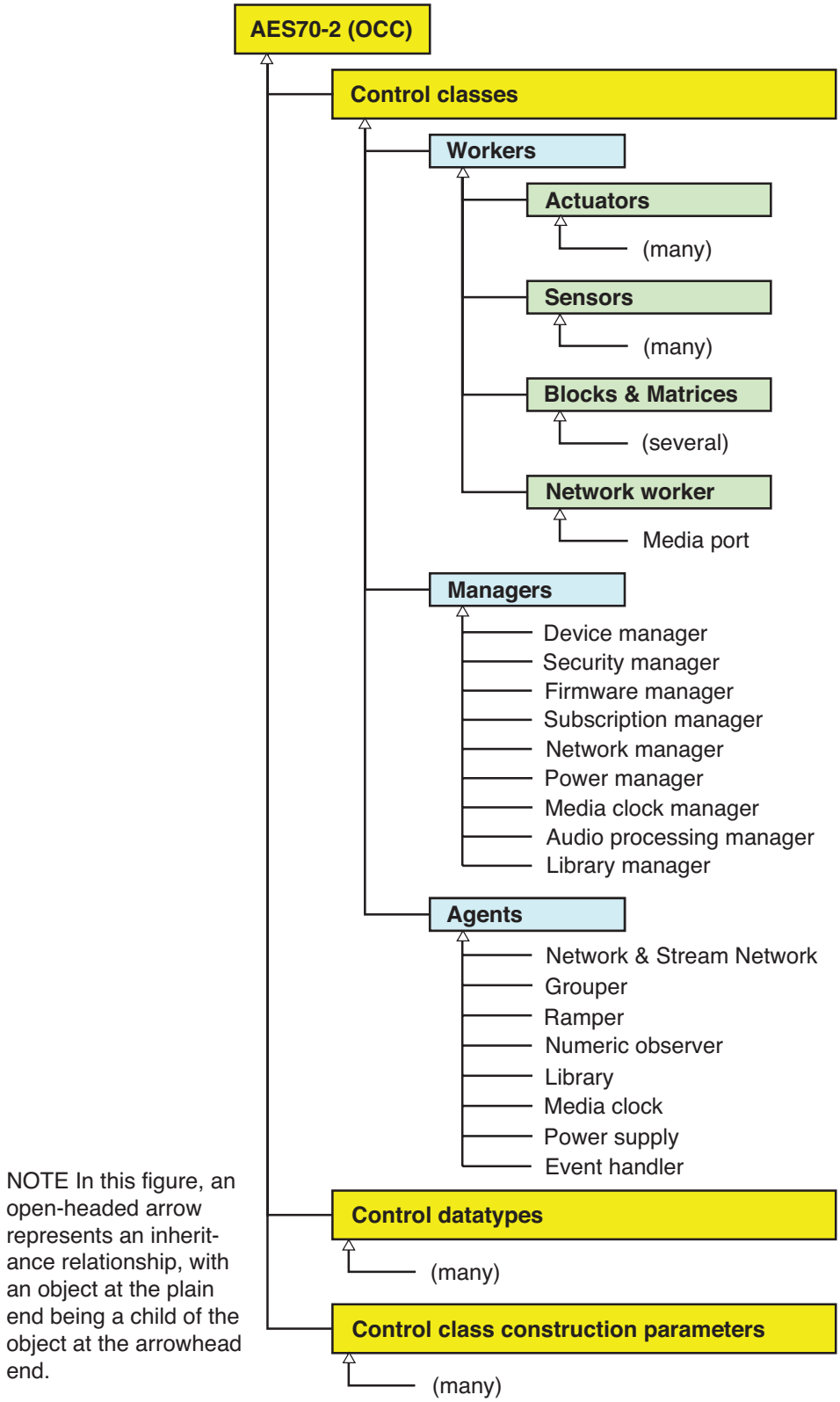
An AES70 device model consists entirely of objects of classes from these categories.

5.3 Datatypes

AES70-2 also provides two sets of supporting definitions:

Control Datatypes	Datatypes used by the control classes.
Control Class Construction Parameters	Datatypes used in the construction of control classes

An overview of the class structure is shown in figure 1. Summaries of the class categories follow.



NOTE In this figure, an open-headed arrow represents an inheritance relationship, with an object at the plain end being a child of the object at the arrowhead end.

Figure 1 - Class structure overview

5.4 Worker classes

Worker classes describe device application functions. There are four kinds of Workers:

Actuators	Signal processing and routing functions.
Sensors	Detectors and monitors of various types; for example, signal level, gain reduction, temperature.
Blocks and Matrices	Classes that aggregate objects into structured collections. Used for modeling and managing structures of complex devices.
Network Worker	Class for connection management of incoming and outgoing media streams.

For any given device, a Worker class may be instantiated as many times as necessary to model the device's functions.

Where necessary, a Worker class may be refined and extended by a manufacturer-specific child class. This is explained further in AES70-1.

5.5 Manager Classes

Manager classes describe device housekeeping functions. For each device, manager classes are singletons; that is, each one is instantiated at most once at most per device.

Some manager classes are required for AES70 compliance and shall be instantiated in every device; others are optional. Not all interface elements of all required classes are required. Minimum device requirements for AES70 compliance are defined in annex B.

The manager classes are:

Device Manager	Contains manufacturer and model information and controls overall device state.
Security Manager	Controls security features, or reports that there are none.
Firmware Manager	Manages device firmware updating, or reports that it is not implemented.
Subscription Manager	Manages the reporting of device data back to controllers.
Network Manager	Holds a collection of the device's network interfaces, as defined by the network objects mentioned in clause 5.6.
Power Manager	Allows control and monitoring of device's power supply or supplies.
Media Clock Manager	Gives access to device's media clocking features.
Audio Processing Manager	Gives access to global parameters controlling audio processing.
Library Manager	Controls creation, management, and use of stored parameter sets (also known as "presets", "patches", and so on)
Device Time Manager	Gives access to the device's time-of-day clock, if any.

5.6 Agent Classes

These are the classes that provide notable control features that are not directly related to signal processing. The agent classes are:

OcaNetwork	Represents connections to control networks.
OcaStreamNetwork	Represents connections to control and/or media networks.
OcaStreamConnector	Represents an **OcaStreamNetwork** connection point for external media streams.
OcaGrouper	Supports control aggregation, allowing a single parameter change to affect many objects. Similar in effect to a VCA master on a mixing console.
OcaRamper	Provides incremental parameter changes - timed fades, for example. Also provides queuing of parameter changes to occur at specified times in the future.
OcaNumericObserver	A "watcher" that observes a particular parameter, and alerts controllers when it reaches a particular value. Also supports periodic reporting of parameter values (for example, level meter readings) to controllers.
OcaNumericObserverList	Same as **OcaNumericObserver**, but observes a list of parameters.
OcaLibrary	Provides a range of functions for pre-storing sets of parameters in the device and applying them when desired.
OcaMediaClock	Describes an internal or external media clock that the device uses. Can be multiply instantiated for devices which support more than one media clock.
OcaEventHandler	Describes the controller interface that handles incoming notifications from controlled devices.

5.7 Control Datatypes

AES70-2 defines a range of control datatypes. These datatypes are used in the definitions of the classes listed above. Details are in annex A.

5.8 Control Class Construction Parameters

Some DSP-based products allow controllers to define their processing topologies. AES70 refers to these as *configurable devices*. In some configurable devices, controllers can cause new processing objects to be created and deleted. AES70 refers to these as *fully-configurable devices***.**

When a controller creates an object in a fully configurable device, certain parameters may be required. For example, if a controller creates a multiposition switch, it needs to specify how many positions the switch has and, optionally, a text label for each position. Such parameters are called *construction parameters*.

The number and kind of construction parameters varies from class to class. AES70-3 contains a separate subtree that specifies the construction parameters that are required for each class. This subtree has an entry corresponding to every Worker class and every Agent class.

AES70 does not support the runtime creation of Manager objects by fully-configurable devices, so these are not included in the construction parameter set.

5.9 Control Class and Element Identification

Schemes for identifying control classes and elements are described in AES70-1 clause 4.1.2.1.

Annex A (normative) - UML Class Structure Definition

The content of the Annex is an external XMI 2.1 document, as described above in clause 4. It may be downloaded from:

www.aes.org/standards/models/AES70-2-AnnexA-151112-class-structure-1.xmi

NOTE: as an informative equivalent, users may like to refer to a proprietary Enterprise Architect version:

www.aes.org/standards/models/AES70-2-AnnexA-151112-class-structure-1.eap

Annex B (normative) - Minimum OCC implementation

B.1 Introduction

This Annex specifies the minimum device model that a device shall implement to be compliant with AES70.

In what follows here, an AES70-compliant device is referred to as an *AES70 device,* and the device model of such a device is referred to as a *compliant device model.*

B.2 AES70 compliance

Every AES70-compliant device shall implement at least the minimum device model elements specified in this Annex, and shall implement at least one AES70 protocol. An AES70 protocol is defined in AES70-3.

B.3 Required objects

B.3.1 General

This section identifies objects that are required for compliance.

A 'minimum implementation' necessarily depends on whether the device is required to support encrypted command streams (*Secure*); or send and receive digital media streams over a network (*Streaming*); or both.

An AES70 device may include optional objects as needed to render some or all of its functions accessible for control and/or monitoring from the connected network.

> NOTE AES70 compliance does not require a device to include AES70 Workers or Agents for all of its functions; manufacturers may freely select which functions to make controllable via the network.

B.3.2 Required Managers

The manager objects that shall be implemented by an AES70 device are shown in table B.1.

Each required object shall implement all of the methods defined by its class. Many of these methods may return a "Not Implemented" status where appropriate. The model identified in clause 4 provides details.

Table B.1 - Required manager objects

Manager Object	ONo	Required for devices:		
		All	Secure	Streaming
`OcaDeviceManager`	1	●		
`OcaSecurityManager`	2		●	
`OcaFirmwareManager`	3	●		
`OcaSubscriptionManager`	5	●		
`OcaNetworkManager`	6	●		
`OcaMediaClockManager`	7			●

B.3.3 Required Workers

The Worker objects that shall be implemented by all AES70 devices are shown in table B.2.

Table B.2 - Required Worker objects

Worker Object	ONo	Required for devices:		
		All	Security	Streaming
OcaBlock	100	●		
OcaMediaClock	varies			●

B.3.4 Required Agent

An AES70 device shall implement at least one Agent, a *network object.* A network object shall be an object of class **OcaStreamNetwork**.

> NOTE earlier OCA implementations may also use an **OcaNetwork** network object, but this is deprecated for new designs.

B.4 Firmware Upgrading

AES70 devices that do not implement the AES70 firmware upgrade feature shall provide a reduced **OcaFirmwareManager** which supplies the version numbers of firmware components in the device. See B.5.5 for details.

B.5 Required Methods and Events for Required Objects

B.5.1 General

All methods that are defined by a required object's class shall be present in the device model. Where a method is not implemented in the particular device it shall return a **NotImplemented** result.

In the text that follows, "all methods and events" refer to all methods and events of the class being described, as specified in clause 4.

B.5.2 Base set

GetLockable(...)	Only read-only objects may return False
Lock(...)	Only if object is lockable
Unlock(...)	Only if object is lockable
event PropertyChanged(...)	

B.5.3 OcaDeviceManager

GetDeviceName(...)	
GetEnabled(...)	
SetEnabled(...)	
GetManagers(...)	
GetModelDescription(...)	
GetModelGUID(...)	

GetOcaVersion(...)	
GetSerialNumber(...)	
GetState(...)	

B.5.4 OcaSecurityManager

When an **OcaSecurityManager** object is implemented, it shall implement all methods and events of the **OcaSecurityManager** class.

B.5.5 OcaFirmwareManager

The **OcaFirmwareManager** object shall implement the **GetComponentVersions(...)** method for all devices.

For devices that use the OCA firmware upgrade feature, the **OcaFirmwareManager** object shall implement all other methods and events of the **OcaFirmwareManager** class.

B.5.6 OcaSubscriptionManager

AddSubscription(...)	
RemoveSubscription(...)	

B.5.7 OcaNetworkManager

The **OcaNetworkManager** object shall implement all methods and events of the **OcaNetworkManager** class.

B.5.8 OcaMediaClockManager

When an **OcaMediaClockManager** object is implemented, it shall implement all methods and events of the **OcaMediaClockManager** class.

B.5.9 OcaBlock

GetEnabled(...)	(inherited from **OcaWorker**)
SetEnabled(...)	(inherited from **OcaWorker**)
GetPorts(...)	(inherited from **OcaWorker**)
GetMembers(...)	
GetMembersRecursive(...)	

B.5.10 OcaStreamNetwork and OcaNetwork

When an **OcaStreamNetwork** object is implemented, it shall implement all methods and events of the **OcaStreamNetwork** class.

Where an **OcaNetwork** object is implemented, it shall implement all methods and events of the **OcaNetwork** class.

NOTE **OcaNetwork** is a deprecated class.

B.5.11 OcaMediaClock

When an **OcaMediaClock** object is implemented, it shall implement all methods and events of the **OcaMediaClock** class.