



# TA Document 1999021

## AV/C Device Presets 1.0

July 10, 2000

**Sponsored by:**

1394 Trade Association

**Accepted for Release by:**

1394 Trade Association Board of Directors.

**Abstract:**

This document defines the AV/C DEVICE PRESET unit command and its associated descriptor structures.

**Keywords:**

AV/C, Presets, Unit Commands.

---

Copyright © 1996-2000 by the 1394 Trade Association.  
Regency Plaza Suite 350, 2350 Mission College Blvd., Santa Clara, CA 95054, USA  
<http://www.1394TA.org>  
All rights reserved.

Permission is granted to members of the 1394 Trade Association to reproduce this document for their own use or the use of other 1394 Trade Association members only, provided this notice is included. All other rights reserved. Duplication for sale, or for commercial or for-profit use is strictly prohibited without the prior written consent of the 1394 Trade Association.

**1394 Trade Association Specifications** are developed within Working Groups of the 1394 Trade Association, a non-profit industry association devoted to the promotion of and growth of the market for IEEE 1394-compliant products. Participants in working groups serve voluntarily and without compensation from the Trade Association. Most participants represent member organizations of the 1394 Trade Association. The specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Specification is wholly voluntary. The existence of a 1394 Trade Association Specification is not meant to imply that there are not other ways to produce, test, measure, purchase, market or provide other goods and services related to the scope of the 1394 Trade Association Specification. Furthermore, the viewpoint expressed at the time a specification is accepted and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Specification.

Comments for revision of 1394 Trade Association Specifications are welcome from any interested party, regardless of membership affiliation with the 1394 Trade Association. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally, questions may arise about the meaning of specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will initiate action to prepare appropriate responses.

Comments on specifications and requests for interpretations should be addressed to:

Editor, 1394 Trade Association  
Regency Plaza Suite 350  
2350 Mission College Blvd.  
Santa Clara, Calif. 95054, USA

1394 Trade Association Specifications are adopted by the 1394 Trade Association without regard to patents which may exist on articles, materials or processes or to other proprietary intellectual property which may exist within a specification. Adoption of a specification by the 1394 Trade Association does not assume any liability to any patent owner or any obligation whatsoever to those parties who rely on the specification documents. Readers of this document are advised to make an independent determination regarding the existence of intellectual property rights, which may be infringed by conformance to this specification.

## Table of Contents

1. Overview .....	6
1.1 Purpose.....	6
1.2 Scope.....	6
2. References .....	7
3. Definitions .....	8
3.1 Conformance Levels .....	8
3.2 Glossary of Terms .....	8
3.3 Acronyms and Abbreviations .....	8
4. Device Preset Model .....	9
4.1 Device Preset categories.....	9
4.1.1 Factory Default Device Preset.....	9
4.1.2 Factory Presets .....	9
4.1.3 User Presets.....	9
4.2 Device Preset Identification .....	9
4.3 Device Preset Discovery .....	10
4.4 Device Preset Actions .....	10
4.4.1 Recall Preset action.....	10
4.4.2 Store Preset action.....	10
4.4.3 Create Preset action.....	10
4.4.4 Delete Preset action.....	10
5. Device Preset Descriptor.....	11
5.1 Device Preset List Descriptor.....	11
5.2 Device Preset Object Entry Descriptor.....	12
6. Device Preset Command .....	13
6.1 DEVICE PRESET control command.....	13
6.1.1 RECALL PRESET subfunction command.....	13
6.1.2 RECALL PRESET subfunction response.....	14
6.1.3 STORE PRESET subfunction command.....	14
6.1.4 STORE PRESET subfunction response.....	14
6.1.5 CREATE PRESET subfunction command.....	14
6.1.6 CREATE PRESET subfunction response.....	14
6.1.7 DELETE PRESET subfunction command.....	15
6.1.8 DELETE PRESET subfunction response.....	15

## List of Figures

Figure 5.1 – Device Preset List Descriptor.....	11
Figure 5.2 – Device Preset Object Entry Descriptor.....	12
Figure 6.1 – DEVICE PRESET general command frame.....	13
Figure 6.2 – RECALL PRESET subfunction frame.....	13
Figure 6.3 – STORE PRESET subfunction frame.....	14
Figure 6.4 – CREATE PRESET subfunction command frame.....	14
Figure 6.5 – CREATE PRESET subfunction response frame.....	14
Figure 6.6 – DELETE PRESET subfunction frame.....	15

## List of Tables

Table 4.1 – Preset_ID range allocation .....	10
Table 5.1 – generation_ID field definition .....	12
Table 6.1 – Support level of DEVICE PRESET command.....	13
Table 6.2 – subfunction field definition .....	13

## 1. Overview

### 1.1 Purpose

The purpose of this document is to define a mechanism for managing saved states of audio/visual devices, called Device Presets.

### 1.2 Scope

The scope of this document is to define the DEVICE PRESET AV/C unit command and its associated descriptor structures.

## 2. References

The following standards contain provisions, which through reference in this document, constitute provisions of this standard. All the standards listed are normative references. Informative references are given in Annex A. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

- [R1] IEEE Std 1394-1995, Standard for a High Performance Serial Bus.
- [R2] IEC-61883-1, Consumer audio/video equipment – Digital interface – Part 1: General.
- [R3] AV/C Digital Interface Command Set General Specification, Version 3.0. TA document number 1998003.
- [R4] Enhancements to the AV/C General Specification 3.0, Version 1.0. TA document number 1998010.

## 3. Definitions

### 3.1 Conformance Levels

**3.1.1 expected:** A key word used to describe the behavior of the hardware or software in the design models *assumed* by this Specification. Other hardware and software design models may also be implemented.

**3.1.2 may:** A key word that indicates flexibility of choice with *no implied preference*.

**3.1.3 shall:** A key word indicating a mandatory requirement. Designers are *required* to implement all such mandatory requirements.

**3.1.4 should:** A key word indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase *is recommended*.

**3.1.5 reserved fields:** A set of bits within a data structure that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall zero these fields. Future revisions of this specification, however, may define their usage.

**3.1.6 reserved values:** A set of values for a field that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall not generate these values for the field. Future revisions of this specification, however, may define their usage.

### 3.2 Glossary of Terms

**3.2.1 byte:** Eight bits of data, used as a synonym for octet.

**3.2.2 CSR Architecture:** A convenient abbreviation of the following reference (see clause 2): ISO/IEC 13213 : 1994 [ANSI/IEEE Std 1212, 1994 Edition], Information Technology—Microprocessor systems—Control and Status Register (CSR) Architecture for Microcomputer Buses.

**3.2.3 Device Preset:** A saved configuration of an audio visual device's parameters appropriate to the device type.

**3.2.4 quadlet:** Four bytes of data.

### 3.3 Acronyms and Abbreviations

AV/C                      Audio Video Control.



## 4. Device Preset Model

### 4.1 Device Preset categories

A Device Preset is a saved configuration of an audio/visual device's parameters, appropriate to the device type. The parameters that are effected by a Device Preset are device and application specific, and beyond the scope of this specification.

There are three Device Preset categories:

- 1) Factory Default Device Preset.
- 2) Factory Device Presets.
- 3) User Device Presets.

#### 4.1.1 Factory Default Device Preset

The *Factory Default Device Preset* is a known safe state of the device's configurable parameters.

#### 4.1.2 Factory Presets

*Factory Device Presets* are pre-configured states of a device's configurable parameters that can be recalled by a user.

Consider the example of an audio receiver. This device may have the *Factory Device Presets* of "JAZZ", "ROCK", and "CLASSICAL", each applying a different set of tone controls.

A display device may have the *Factory Device Presets* "TELEVISION MODE", and "MOVIE MODE", each applying different aspect ratio and brightness settings.

#### 4.1.3 User Presets

*User Device Presets* are user-configured states of a device's configurable parameters. Typically, a user stores a User Device Preset after adjusting the device's settings.

Using the audio receiver example again, a user may adjust the tone controls and store a "MY JAZZ" *User Device Preset*.

### 4.2 Device Preset Identification

Device Presets are uniquely identified within a device by a *Preset\_ID*. The range of *Preset\_ID* values is divided into ranges for the Device Preset categories, as shown below in Table 4.1.

**Table 4.1 – Preset\_ID range allocation**

<b>Range of values</b>	<b>Device Preset category</b>
00 00 <sub>16</sub>	Factory Default Device Preset
00 01 <sub>16</sub> to 00 FF <sub>16</sub>	Factory Device Presets
01 00 <sub>16</sub> to 01 FF <sub>16</sub>	User Device Presets
02 00 <sub>16</sub> to FF FF <sub>16</sub>	Reserved for future definition

### 4.3 Device Preset Discovery

Details of Device Presets that are available in a device are stored in an AV/C descriptor structure called the Device Preset List Descriptor. Details provided for each Device Preset include a Preset\_ID and a textual description.

The Device Preset List Descriptor is a root list accessible via the unit descriptor. Details of this descriptor are provided in clause 5.

### 4.4 Device Preset Actions

Device Preset actions are initiated by the DEVICE PRESET unit command. Details of this command are provided in clause 6. There are four types of Device Preset actions defined:

- 1) Recall Preset
- 2) Store Preset
- 3) Create Preset
- 4) Delete Preset

#### 4.4.1 Recall Preset action

The Recall Preset action is used to set a device's configurable parameters to a previously saved state. This action is applicable to all Device Preset categories.

#### 4.4.2 Store Preset action

The Store Preset action is used to save a device's configurable parameters as a Device Preset. The Device Preset specified must be an existing User Device Preset.

#### 4.4.3 Create Preset action

The Create Preset action is used to create a new Device Preset within the User Device Preset category, into which the device's configurable parameters are saved.

#### 4.4.4 Delete Preset action

The Delete Preset action is used to delete the specified User Device Preset.

## 5. Device Preset Descriptor

### 5.1 Device Preset List Descriptor

A device that implements the AV/C DEVICE PRESET command shall maintain a `device_preset_list`. The `device_preset_list` is a `root_object_list` that is accessible via the unit descriptor.

The structure of the `device_preset_list` descriptor is defined below in Figure 5.1.

Address	Length, bytes	External Read/Write	Static/Dynamic	Contents
00 00 <sub>16</sub>	2	R	D	Descriptor_length
00 01 <sub>16</sub>				
00 02 <sub>16</sub>	1	R	S	List_type = 0 (DEVICE PRESET LIST)
00 03 <sub>16</sub>				Attributes
00 04 <sub>16</sub>	2	R	S	Size_of_list_specific_information = 00 02
00 05 <sub>16</sub>				
00 06 <sub>16</sub>	1	R	S	Device_Presets_Generation_ID
00 07 <sub>16</sub>	1	R	S	Max_number_of_entries
00 08 <sub>16</sub>	2	R	D	Number_of_entries (n)
00 09 <sub>16</sub>				
00 0A <sub>16</sub>	-	-	-	Device_preset_object_entry [0]
:				
:				
:				
:	-	-	-	Device_preset_object_entry[n-1]
:				
:				
:				

Figure 5.1 – Device Preset List Descriptor.

Refer to [R3] for descriptions of `descriptor_length`, `list_type`, `attributes`, `size_of_specific_information`, and `number_of_entries`.

The value of `list_type` for the Device Preset List Descriptor shall be 0.

The value of the `size_of_specific_information` field shall be 00 02<sub>16</sub>.

The value of the `Device_Presets_Generation_ID` field determines the version of the Device Preset command and descriptor structure that the device supports. The possible values are defined in Table 5.1 below.

**Table 5.1 – generation\_ID field definition**

Value	Meaning
00 <sub>16</sub>	Device implements version 1.0 of AV/C Device Presets.
All other values.	Reserved for future definition.

The value of the *max\_number\_of\_entries* field determines the total number of User Device Presets supported.

## 5.2 Device Preset Object Entry Descriptor

The device\_preset\_object\_entry descriptor is defined in Figure 5.2 below.

Address	Length, bytes	External Read/Write	Static/Dynamic	Contents
00 00 <sub>16</sub>	2	R	D	Descriptor length
00 01 <sub>16</sub>				
00 02 <sub>16</sub>	1	R	S	Entry_type = 0
00 03 <sub>16</sub>	1	R	D	Attributes
00 04 <sub>16</sub>	N	R	S	Object_ID
:				
:				
:	2	R	S	Size_of_entry_specific_information
:				
:	M	W	D	Description_info_block
:				
:				

**Figure 5.2 – Device Preset Object Entry Descriptor.**

Refer to [R3] for a description of *descriptor\_length*.

The *entry\_type* field shall be the 0, indicating that the entry is a Device Preset Object Entry descriptor.

The *attributes* field shall have the *has\_child\_ID* bit set to 0. Refer to [R3] for values of other bits.

The least significant two bytes of the *object\_ID* field shall indicate the *preset\_ID*. Note that the size of the *object\_ID* field is specified within the unit descriptor.

The value of the *size\_of\_entry\_specific\_information* shall be 0.

Refer to [R4] for the definition of the *description\_info\_block*.

## 6. Device Preset Command

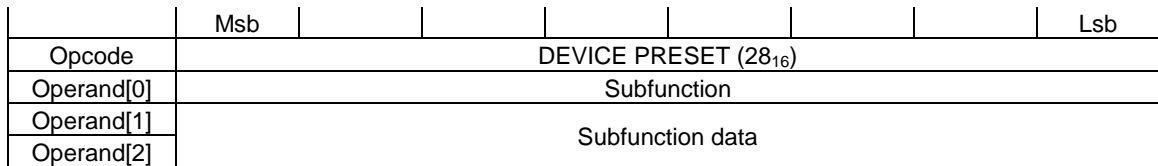
The support level for the DEVICE PRESET unit command is summarized in Table 6.1 below:

**Table 6.1 – Support level of DEVICE PRESET command**

Opcode	Value	Support level (by ctype)			Comments
		C	S	N	
DEVICE-PRESET	28 <sub>16</sub>	O	-	-	Manage device presets

### 6.1 DEVICE PRESET control command.

The general format of the DEVICE PRESET control command is as follows:



**Figure 6.1 – DEVICE PRESET general command frame.**

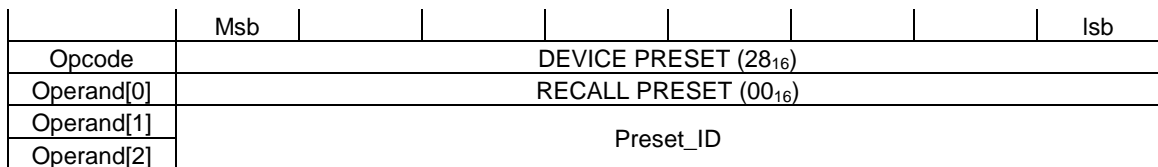
The *subfunction* field indicates the actions to be taken in the target, for command frames with a ctype of *control*, as defined below in Table 6.2.

**Table 6.2 – subfunction field definition**

Symbol	Value	Meaning
RECALL PRESET	00 <sub>16</sub>	Recall a device preset.
STORE PRESET	01 <sub>16</sub>	Store current state of device as a preset.
CREATE PRESET	02 <sub>16</sub>	Create a new user preset.
DELETE PRESET	03 <sub>16</sub>	Delete a user preset.
	All other values	Reserved.

#### 6.1.1 RECALL PRESET subfunction command.

The format of the RECALL PRESET subfunction control command is defined below in Figure 6.2.



**Figure 6.2 – RECALL PRESET subfunction frame.**

The *Preset\_ID* shall indicate the device state to restore.

### 6.1.2 RECALL PRESET subfunction response.

The format of the RECALL PRESET subfunction response frame is the same as the control frame, shown above in Figure 6.2.

### 6.1.3 STORE PRESET subfunction command.

The format of the STORE PRESET subfunction control command is defined below in Figure 6.3.

	MsB							lsB
Opcode	DEVICE PRESET (28 <sub>16</sub> )							
Operand[0]	STORE PRESET (01 <sub>16</sub> )							
Operand[1]	Preset_ID							
Operand[2]								

**Figure 6.3 – STORE PRESET subfunction frame.**

The *Preset\_ID* shall indicate an existing user Preset\_ID within the *User Presets* range, with which the current device state will be associated.

### 6.1.4 STORE PRESET subfunction response.

The format of the STORE PRESET subfunction response frame is the same as the control frame, shown above in Figure 6.3.

### 6.1.5 CREATE PRESET subfunction command.

The format of the CREATE PRESET subfunction control command is defined below in Figure 6.4.

	MsB							lsB
Opcode	DEVICE PRESET (28 <sub>16</sub> )							
Operand[0]	CREATE PRESET (02 <sub>16</sub> )							
Operand[1]	FF FF <sub>16</sub>							
Operand[2]								

**Figure 6.4 – CREATE PRESET subfunction command frame.**

### 6.1.6 CREATE PRESET subfunction response.

The format of the CREATE PRESET subfunction response frame is defined below in Figure 6.4.

	MsB							lsB
Opcode	DEVICE PRESET (28 <sub>16</sub> )							
Operand[0]	CREATE PRESET (02 <sub>16</sub> )							
Operand[1]	Preset_ID							
Operand[2]								

**Figure 6.5 – CREATE PRESET subfunction response frame.**

The *Preset\_ID* indicates a new Preset\_ID within the *User Presets* range that the current state of the device has been associated with.

Before completion of the CREATE PRESET subfunction, a new *device\_preset\_object\_entry* shall be added to the *device\_preset\_list* descriptor.

If the new CREATE PRESET subfunction is initiated by a controller in another unit, the *attribute* field of the newly created *device\_preset\_object\_entry* descriptor shall have the *skip* bit set until the controller writes the *description\_info\_block* for the object entry.

### 6.1.7 DELETE PRESET subfunction command.

The format of the DELETE PRESET subfunction control command is defined below in Figure 6.6.

	MsB						lsB
Opcode	DEVICE PRESET (28 <sub>16</sub> )						
Operand[0]	DELETE_PRESET (03 <sub>16</sub> )						
Operand[1]	Preset_ID						
Operand[2]							

**Figure 6.6 – DELETE PRESET subfunction frame.**

The *Preset\_ID* shall indicate an existing user *Preset\_ID* within the *User Presets* range that is to be deleted.

### 6.1.8 DELETE PRESET subfunction response.

The format of the DELETE PRESET subfunction response frame is the same as the control frame, shown above in Figure 6.6.

Before completion of the DELETE PRESET subfunction, the associated *device\_preset\_object\_entry* shall be deleted from the *device\_preset\_list* descriptor.