# SMPTE STANDARD

# D-Cinema Operations — Auxiliary Content Synchronization Protocol



## Table of Contents





Approved
November 2, 2010

## Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in Part XIII of its Administrative Practices.

SMPTE ST 430-10 was prepared by Technology Committee 21DC.

## Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

# 1 Scope

This document defines a protocol for synchronizing auxiliary resources in a composition playlist to the playback timeline.

# 2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

# 3 Normative References

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 336:2007) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 336M-2007). Documents with the same root number (e.g. 336) and publication year (e.g. 2007) are functionally identical.

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. 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 edition of the standards indicated below.

[336] SMPTE ST 336:2007, Data Encoding Protocol Using Key-Length-Value

[IANA] Internet Assigned Numbers Authority. See www.iana.org/assignments/port-numbers

[CPL] SMPTE ST 429-7:2006, D-Cinema Packaging — Composition Playlist

[RPL] SMPTE ST 430-11:2010, D-Cinema Operations — Auxiliary Resource Presentation List

# 4 Glossary

The following paragraphs define the acronyms used in this document.

**ACS:** Auxiliary Content Server
**BER:** Basic Encoding Rules
**CSP:** Auxiliary Content Synchronization Protocol (this document)
**DCS:** Digital Cinema Server
**HTTP:** Hypertext Transfer Protocol
**KLV:** Key Length Value
**RRP:** Request-Response Pair
**Show:** An ordered sequence of compositions
**RPL:** Auxiliary Resource Presentation List
**TCP:** Transmission Control Protocol
**Uint8:** 8-bit unsigned integer
**Uint32:** 32-bit unsigned integer
**Uint64:** 64-bit unsigned integer
**Int64:** 64-bit signed integer
**UUID:** Universally Unique Identifier
**URI:** Uniform Resource Identifier
**URL:** Uniform Resource Locator

# 5 Overview (Informative)

This document defines a protocol for synchronizing auxiliary resources in an Auxiliary Resource Presentation List (RPL) to the playback timeline. This protocol is intended for use between a Digital Cinema Server (DCS) and one or more Auxiliary Content Servers (ACS). Examples of such servers are:

- Caption servers
- Special effects servers (e.g. lightning strobe, chair rumble)
- Secondary display servers

## 5.1 Composition Playlist Resources

The Composition Playlist defines the resources to be presented during playout. At a minimum, these resources will consist of video and/or audio but may also contain subtitles, several types of captions, and other possible tracks. While each resource track is referred to by a single ID, the actual resource may consist of many files; e.g. font file, XML files, image files, etc.

### 5.2 Resource Loading

While the audio and video tracks in the Composition Playlist can be streamed in real time on a frame-by-frame basis, auxiliary content resources may have files that need to be preloaded (e.g. font files). Auxiliary content resources may be preloaded into the ACS to ensure that all concurrent files are ready to be used.

### 5.3 Protocol Description

Prior to playout, the DCS will create a new Auxiliary Resource Presentation list (RPL) file, whose location will be sent to the ACS. The ACS will use HTTP to connect back to the DCS to acquire the RPL and the first set of resources that needs to be cached. After the ACS has finished caching resources, it will indicate its status as OK. Once playback starts, the DCS will enable resource output and send timeline updates to the ACS which will render the resource associated with the current timeline value.

## 6 Communications Protocol

### 6.1 Control Protocol

The Control Protocol is a TCP-based protocol used to instruct the ACS which resources should be played with the timeline and when to play the referenced resources. The protocol operates in a master/slave fashion with the DCS sending requests and the ACS sending synchronous responses. As outlined in Annex B, the connection is first initiated by the ACS.

The SMPTE standardized messages in this protocol shall use well-known port 4170 which has been reserved for D-Cinema Auxiliary Content Synchronization Protocol RRPs by the Internet Assigned Numbers Authority (see [IANA]).

#### 6.1.1 Connection Establishment

The DCS shall listen for a connection on TCP port 4170. On connection initiation, the DCS must send the Announce request (See 7.2.1) to the ACS. On successful receipt of the Announce response, the DCS may send subsequent requests depending on whether resources need to be presented for the next playout.

- The DCS may send no future messages past the Announce request to the ACS if the next show or composition does not require auxiliary resources.
- The DCS shall send the Announce message regardless of whether communications to the ACS is necessary for the next playout.

#### 6.1.2 Message Structure: Key Length Value (KLV)

Request and Response messages shall be Key Length Value (KLV) encoded using Fixed Length Pack encoding according to SMPTE ST 336 [336]. The Fixed Length Universal Label (UL) Key is given in Annex A of this document. As a Fixed Length Pack, each individual item in the Value field comprises only an item value. The KLV Length field shall be a long-form BER value encoded with a fixed length of 4 bytes total.

*Example*: For a KLV packet having a Value field that is 12 bytes in length, the Length field would be encoded as the following 4 bytes, 0x83 0x00 0x00 0x0C (hexadecimal).

Each Request-Response Pair (RRP) represents two message types and thus KLV UL “value” registration is required twice for each defined RRP (see Annex A ).

### 6.2 General Message Elements

For each message type, the following shall apply:

- The message type is denoted within the opening KLV "Key" field (16 bytes).
- "Length" is a BER-encoded four byte field which describes the length of the message in bytes.
- "Request_ID" shall be an application level tag for the Request, which shall be echoed in the corresponding Response. A non-zero Request_ID value shall be set by the DCS, which should select unique values (e.g. a sequencing counter) for each connection it manages. (Request_ID generation means is left to implementers and is out of the scope of this specification.)
- Multi-byte integer values shall be sent as big-endian data; i.e., the most significant byte is transmitted first.

### 6.3 General Message Requirements

RRP transactions shall be synchronous (i.e., each pairing shall be opened and closed before a new RRP is opened between the same two entities). To avoid hang-ups, ACS implementations shall have a maximum round-trip Request-to-Response latency of 2 seconds. Latency shall be measured from the end of the "Request" message receipt to the start of the "Response" message transmission. If the ACS is too busy to accept or process a request, it shall send the corresponding response with the General Status Response Key of the Status Response set to ACS Busy.

Should the ACS fail to respond (at all) after the 2 second time limit, the DCS may consider this a communications failure condition and may close the TCP connection.

### 6.4 Status Responses

Each response message shall contain a general status response value. The Status Response element is represented by the following structure:

| Item Name | Type | Len | Meaning |
|---|---|---|---|
| General Status Response Key | Uint8 | 1 | General Status Response |
| Response Text Length | BER Length | 4 | Response Text Length |
| Response Text Value | Text | Var | Response Text Value |

#### 6.4.1 General Status Response Key

The General Status Response element identifies the general response for the response message. Possible response values are as follows:

| Element | Meaning | UINT8 Value |
|---|---|---|
| RRP successful | Request successfully processed | 0 |
| RRP failed | ACS unable to process Request | 1 |
| RRP Invalid | Invalid parameter or Request structure | 2 |
| ACS Busy | ACS too busy to process Request | 3 |
| Lease Timeout | Lease not valid for last transaction | 4 |
| Playout ID Mismatch | RPL and CSP Playout ID mismatch | 5 |
| General Error | General Error response | 6 |
| Recoverable Error | Recoverable Error | 7 |
| RPL Error | Error parsing RPL | 8 |
| Resource Error | Error parsing resource file | 9 |
| Processing | ACS busy getting or interpreting resources | 10 |
| Reserved | Reserved Response Elements | 11-255 |

Note: The Recoverable Error response can be returned by the ACS to indicate a non-critical error that occurred during operation. This error can be ignored by the DCS but can also be used for logging purposes.

### 6.4.2 Response Text Length

The Response Text Length field defines the length of the Response Text Value. If no Response Text Value is provided, the Response Text Length shall be set to the BER-encoded value of 0.

### 6.4.3 Response Text Value

The Response Text Value field is a human-readable text value providing details about the associated General Status Response. The Response Text Value field shall be encoded using UTF-8 character encoding. If no Text Value is provided, this portion of the response is not sent.

## 7 General Purpose Messages

This section defines the general messages used in the protocol.

### 7.1 Bad Request Response

Each RRP Response message contains a general "Status Response" element; however instances may arise where the ACS does not understand the incident Request that was received. In such case the appropriate Response would be unknown. The Bad Request Response shall be used when the ACS cannot otherwise respond with a Response appropriate to the incident Request. A complete copy of the Request message as received by the ACS is carried in this message.

*Bad Request Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Bad Request Response UL Name | Pack Key | 16 | | Identifies the Bad Request Response |
| Length | BER Length | 4 | n/a | Pack length |
| Request Copy Length | BER Length | 4 | n/a | Request Copy Length |
| Request Copy | Text | Var | | Copy of Request |
| Status Response | KLV | Var | | Status Response |

- The Request Copy is a complete copy of the Request message that was not understood.

## 7.2 Message Summary

**Table 1 – Message Summary**

| Message | Description |
|---|---|
| Announce | Initial announcement |
| Get New Lease | Establish a new connection lease |
| Get Status | Acquire high-level status of the ACS |
| Set RPL Location | Specify the HTTP URL for the resource map file |
| Set Output Mode | Instruct ACS to enable or disable resource output |
| Update Timeline | Update the current edit unit being displayed |
| Terminate Lease | Terminate the connection lease |
| Get Log Event List | Acquire the list of Log Events for a time range |
| Get Log Event | Acquire a Log Event |

### 7.2.1 Announce

After the ACS initiated the connection with the DCS, the Announce request is sent by the DCS to the ACS. This message is used to provide identification of both sides of the communication and to ensure that the recipient device is capable of understanding the protocol.

*Announce Request*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Announce Request UL Name | Pack Key | 16 | | Identifies the Announce Request |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | ID of this Request |
| Current Time | Int64 | 8 | | Current system time in seconds from epoch |
| Device Description | Text | Var | | Human-Readable DCS device description |

*Announce Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Announce Response UL Name | Pack Key | 16 | | Identifies the Announce Response |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | IDs the Request for which this is the Response |
| Current Time | Int64 | 8 | | Current system time in seconds from epoch |
| Device Description Length | BER Length | 4 | | Length of text in Device Description field |
| Device Description | Text | Var | | Human-Readable ACS device description |
| Status Response | KLV | Var | | Status Response Element |

#### 7.2.1.1 Current Time

The Current Time parameter is a UTC based time value provided by both the DCS on request and ACS on response. This time can be used by each device to correlate the current time on the other device for logging purposes.

#### 7.2.1.2 Device Description

The Device Description field is a human-readable string describing the DCS (Announce Request) or ACS (Announce Response). The Device Description field shall be encoded using UTF-8 character encoding.

Note: While the formatting of the device description field is not defined, recommended fields within the device description include vendor name, device model number, and firmware version.

### 7.2.2 Get New Lease

The Get New Lease request is sent by the DCS to the ACS to set the lease duration.

*Get New Lease Request*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Get New Lease Request UL Name | Pack Key | 16 | | Identifies the Get New Lease Request |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | ID of this Request |
| Lease duration | Uint32 | 4 | | Lease duration in seconds |

*Get New Lease Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Get New Lease Response UL Name | Pack Key | 16 | | Identifies the Get New Lease Response |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | IDs the Request for which this is the Response |
| Status Response | KLV | var | | Status Response |

A lease is used to ensure that both sides of the communications channel are alive. The DCS shall send a Get New Lease request prior to sending any request other than Announce. After the response to this request is received, the DCS must send a subsequent request before the lease has expired to keep the lease alive. The lease start time is defined as the time when the ACS sends the Get New Lease response to the DCS. The lease timeout is reset each time any request is received by the ACS. If the lease expires, the ACS should purge all messages and data associated with the lease period and cease resource output.

*Note:* To meet lease timing, the DCS can send a Get New Lease(<lease duration>) request and subsequently send another request, such as Get Status or Update Timeline, at a period of <lease duration>/2.

### 7.2.3 Get Status

The Get Status request obtains the current status of the ACS. This request can be also be used as a lease renewal message between the DCS and the ACS, since the lease timeout is reset each time a request is received (see 7.2.2).

The ACS shall respond to the Get Status request with the current state of the system.

*Get Status Request*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Get Status Request UL Name | Pack Key | 16 | | Identifies the Get Status Request |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | ID of this Request |

*Get Status Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Get Status Response UL Name | Pack Key | 16 | | Identifies the Get Status Response |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | IDs the Request for which this is the Response |
| Status Response | KLV | var | | Status Response |

### 7.2.4 Set RPL Location

The Set RPL Location request provides the URL for the Auxiliary Resource Presentation List. The URL shall be either absolute (including the HTTP protocol and IP address of the RPL) or relative (the ACS shall assume the HTTP protocol and uses the IP address of the DCS as the IP address for the host portion of the URL).

The ACS should verify that the Playout ID sent in the request matches the Playout ID listed in the referenced RPL. If the Playout ID does not match or if there are other errors with the associated RPL, the response value should be set to RPL Error.

If the ACS is not ready to accept the Set RPL Location request, it shall return the Set RPL Location response with a status key of ACS Busy. The DCS shall send the Set RPL Location request until it receives an RRP Successful (see 6.4.1) response.

The DCS may send a Get Status (see 7.2.3) request to determine if the ACS has processed the resources referenced in the Set RPL Location request. If the ACS has not finished processing these resources, it shall return a Processing (see 6.4.1) response. If the ACS has finished processing all resources, it shall return an RRP Successful (see 6.4.1) response.

#### 7.2.4.1 Multiple RPLs

Multiple RPLs may be loaded into an ACS within the same lease. For example, 2 RPLs may be loaded by the DCS representing the current composition/show and the next composition/show.

- The Playout ID shall differ between each RPL.

Note: No requirement is placed on the ACS as to how many RPLs must be supported, but it is recommended that the ACS be able to support at least 2 RPLs to handle the scenario mentioned above.

*Set RPL Location Request*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Set RPL Location Request UL Name | Pack Key | 16 | | Identifies the Set RPL Location Request |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | ID of this Request |
| Playout ID | Uint32 | 4 | | Unique Identifier for the playout to which this RPL applies |
| Resource URL | URL | Var | | URL of the RPL File |

*Set RPL Location Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Set RPL Location Response UL Name | Pack Key | 16 | | Identifies the Set RPL Location Response |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | IDs the Request for which this is the Response |
| Status Response | KLV | Var | | Status Response |

### 7.2.5 Set Output Mode

The Set Output Mode request instructs the ACS to enable or disable resource output. The DCS shall set the output mode to enabled when playing out content that has synchronous auxiliary content. The DCS shall set the output mode to disabled when the playout of content with synchronous auxiliary content is stopped. On receipt of a Set Output Mode Enabled message the ACS shall immediately output resources based on the last Update Timeline message received.

The ACS should preserve all cached resources if the output mode is set to disabled.

*Set Output Mode Request*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Set Output Mode Request UL Name | Pack Key | 16 | | Identifies the Set Output Mode Request |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | ID of this Request |
| Output Mode | Boolean | 1 | | 0 = disabled, 1 = enabled |

*Set Output Mode Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Set Output Mode Response UL Name | Pack Key | 16 | | Identifies the Set Output Mode Response |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | IDs the Request for which this is the Response |
| Status Response | KLV | Var | | Status Response |

### 7.2.6 Update Timeline

The Update Timeline request provides the current position in the show-relative or composition-relative timeline, depending if the associated RPL represents a show or a composition. Each Update Timeline request shall contain (at a minimum) respectively a show-relative position or a composition-relative position indicating the current edit unit displayed.

The ACS shall use the Playout ID to determine the appropriate RPL to use. Only resources associated with the Playout ID shall be used. The ACS shall use Timeline Position to determine its current output based on the RPL and referenced resources. The ACS shall use Edit Rate to determine the speed at which the timeline advances between timeline updates. The timeline does not advance if the Output Mode is set to disabled. An Update Timeline request shall be sent at least once prior to setting Output Mode to Enabled (see 7.2.5).

If the ACS does not have the resources ready for output but does have the RPL URL (see 7.2.4), it shall return an Update Timeline response with the General Status Response Key of the Status Response set to Processing. If the ACS does not have the resources ready for output and does not have the RPL URL (see 7.2.4), it shall return an Update Timeline response with the General Status Response Key of the Status Response set the Playout ID Mismatch.

The Update Timeline message should be sent to the ACS at a minimum frequency of once per minute.

*Update Timeline Request*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Set Output Mode Request UL Name | Pack Key | 16 | | Identifies the UpdateTimeline Request |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | ID of this Request |
| Playout ID | Uint32 | 4 | | Unique identifier for this playout |
| Timeline Position | Uint64 | 8 | | Current edit unit displayed |
| Edit Rate (numerator) | Uint64 | 8 | | Edit Rate Numerator |
| Edit Rate (denominator) | Uint64 | 8 | | Edit Rate Denominator |
| Timeline Extension Count | Uint32 | 4 | | Number of timeline extensions |
| Timeline Extensions | KLV | Var | | Timeline Extension Parameters |

*Update Timeline Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Set Output Mode Response UL Name | Pack Key | 16 | | Identifies the Set Output Mode Response |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | IDs the Request for which this is the Response |
| Status Response | KLV | Var | | Status Response |

#### 7.2.6.1 Playout ID

Playout ID associates the current playout with a specific RPL and its associated resources. The Playout ID shall be regenerated at every playout as part of a new RPL. It is not required to be universally unique.

Upon detecting a change in the Playout ID, the ACS shall cease output of resources pertaining to the last active RPL. The ACS shall resume output when an active RPL and Timeline Playout ID match. If the Playout ID in the Update Timeline request does not match any active RPL, the ACS should cease resource output.

#### 7.2.6.2 Timeline Position

The Timeline Position field defines the number of edit units displayed since the beginning of the playout of the active show or composition, depending if the associated RPL represents a show or a composition.

*Note:* In cases where the RPL represents a show (instead of a composition), the Timeline Position can not be directly converted from edit units into time without taking into account the edit rates of all prior compositions within the RPL.

#### 7.2.6.3 Edit Rate

The Edit Rate defines the number of edit units displayed per second for the current resource. It is expressed as a rational number.

The ACS shall use Edit Rate to determine the speed at which the timeline advances between timeline updates.

#### 7.2.6.4 Timeline Extension Count

The Timeline Extension Count indicates the number of (optional) timeline extensions. This field shall be zero if no Timeline Extensions are included in this request.

#### 7.2.6.5 Timeline Extensions

Timeline extensions are intended to optionally provide more detailed information about the current position on the timeline. Each extension consists of an embedded Key-Length-Value triplet containing the identifier for the extension, the length of the extension value and the extension value. The order of the timeline extensions within the message is not defined.

**Timeline Extension Format**

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Key | Uint32 | 4 | | Timeline Extension identifier |
| Length | BER Length | 4 | n/a | Pack length |
| Value | Var | Var | | Variable length timeline extension value |

**Timeline Extensions**

| Key Name | Key Value | Len | Value Type | Meaning |
|---|---|---|---|---|
| Current Composition Playlist ID | 0 | 16 | UUID | UUID of current composition |
| Current Composition Playlist Position | 1 | 8 | Uint64 | Position within current composition |
| Current Reel ID | 2 | 16 | UUID | UUID of current reel |
| Current Reel Position | 3 | 8 | Uint64 | Position within the current reel |
| Next Composition Playlist ID | 4 | 16 | UUID | UUID of next composition |
| Next Reel ID | 5 | 16 | UUID | UUID of next reel |
| Reserved | 6-255 | Var | Var | Reserved Timeline Extensions |
| User defined Extensions | >256 | Var | Var | User defined Timeline Extensions |

#### 7.2.6.6 Timeline event handling

- The ACS shall continue outputting resources on loss of communications with the DCS until the lease expires.

- The ACS shall allow for discontinuities in the timeline. The timeline, as generated by the DCS, may jump backward or forward to, for example, loop playback or exit a loop. The ACS shall attempt to output resources corresponding to the current timeline position through such discontinuities. Extremely long jumps may require the ACS to acquire or reacquire the resources from the DCS.

### 7.2.7 Terminate Lease

The Terminate Lease request instructs the ACS to end the lease and purge all data associated with the current lease.

*Terminate Lease Request*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Terminate Lease Request UL Name | Pack Key | 16 | | Identifies the Terminate Lease Request |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | ID of this Request |

*Terminate Lease Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Terminate Lease Response UL Name | Pack Key | 16 | | Identifies the Terminate Lease Response |
| Length | BER Length | 4 | n/a | Pack Length |
| Request ID | Uint32 | 4 | | IDs the Request for which this is the Response |
| Status Response | KLV | Var | | Status Response |

### 7.2.8 Get Log Event List

The Get Log Event List request identifies a UTC Time Start and Time Stop period for which the ACS shall respond with a list of identified logged event records it holds. The Time Start and Time Stop elements define a UTC based time window for which logged event information is being requested.

*Get Log Event List Request*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Get Log Event List Request UL Name | Pack Key | 16 | | Identifies the Get Log Event List Request |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | ID of this Request |
| Time Start | Int64 | 8 | | Log Event list period start time |
| Time Stop | Int64 | 8 | | Log Event list period stop time |

*Get Log Event List Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Get Log Event List Response UL Name | Pack Key | 16 | | Identifies the Get Log Event List Response |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | IDs the Request for which this is the Response |
| Event ID Batch | Event ID Batch | 8+4n | | An unordered Batch of Event ID values (see table below) |
| Status Response | KLV | var | | Status Response |

Event ID Batch

| Item Name | Type | Len | UL | Meaning | Default |
|---|---|---|---|---|---|
| Number of Items | Uint32 | 4 | n/a | The number of Items in the Batch | n |
| Item Length | BER Length | 4 | n/a | The length of each Item | 4 |
| Event ID | Uint32 | 4 | | Unique event identifier(s) | |

- The Event ID Batch is a batch of Event IDs. These Event IDs are in no particular order.There shall be only one entry for each Event ID in the Event ID Batch.
- The length of the variable portion of Event ID Batch can be determined by multiplying the Number of Items by the Item Length.
- If no Events are found within the time window, the Number of Items should be set to 0.

  Note: If an ACS does not support logging functions, it can return 0 for this request.
- The batch format returns the Event ID(s) for all the events recorded with time stamps within the specified Request time window (including Time Start and Time Stop).
- Although the Event IDs need not be ordered, the corresponding Log Events should be returned in chronological order.

### 7.2.9 Get Log Event

The Get Log Event request is used to request specific log event records by Event ID (from the list returned in the Get Log Event List Response). The Event ID identifies a specific log record. The ACS returns the requested log record.

*Get Log Event Request*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Get Log Event Request UL Name | Pack Key | 16 | | Identifies the Get Log Event Request |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | ID of this Request |
| Event ID | Uint32 | 4 | | ID of the requested event |

*Get Log Event Response*

| Item Name | Type | Len | UL | Meaning |
|---|---|---|---|---|
| Get Log Event Response UL Name | Pack Key | 16 | | Identifies the Get Log Event Response |
| Length | BER Length | 4 | n/a | Pack length |
| Request ID | Uint32 | 4 | | IDs the Request for which this is the Response |
| Log Event Text Length | BER Length | 4 | | Log Event Text Length |
| Log Event Text | Text | Var | | Log Event being delivered |
| Status Response | KLV | Var | | Status Response |

- Log Event Text is the logged information for the specified event. The structure of the Log Event Text is not defined.
- The length of the Log Record item can be determined from the Length of the message.
- If no Log Event exists for the requested Event ID, the Log Event Text Length should be set to 0.

# Annex A Auxiliary Content Synchronization Protocol Variable Length Universal Label (UL) Key (Normative)

As a Fixed Length Pack (group of KLV elements), each individual item in the Value field comprises only an item value. All items in a pack are required.

**Table A.1 – Common UL Key Value for all CSP Messages**

| Byte No. | Description | Value (hex) | Meaning |
|---|---|---|---|
| 1 | Object Identifier | 06h | Object ID |
| 2 | Label size | 0Eh | Length of UL |
| 3 | Designator | 2Bh | Sub Identifier |
| 4 | Designator | 34h | SMPTE Identifier |
| 5 | Registry Category Designator | 02h | KLV Groups (Sets and Packs) |
| 6 | Registry Designator | 05h | Fixed Length Pack |
| 7 | Structure Designator | 01h | Groups Dictionary |
| 8 | Version Number | 01h | Registry Version: Dictionary version 1 |
| 9 | Item Designator | 02h | Administration |
| 10 | Organization | 07h | Access Control |
| 11 | Application | 02h | Auxiliary Content Synchronization Protocol |
| 12 | Set/Pack Kind (1) | xx | Message Type (see Table A.2) |
| 13 | Set/Pack Kind (2) | yy | Message Type (see Table A.2) |
| 14 | Reserved | 00h | Not assigned |
| 15 | Reserved | 00h | Not assigned |
| 16 | Reserved | 00h | Not assigned |

The values of bytes 12 and 13 for specified CSP message types are given in Table 2:

**Table A.2 – Key Values for Message Types**

| SMPTE UL Name Message Type and byte 13 node names | Byte 12 | Byte 13 |
|---|---|---|
| Bad Request Response | 01h | 01h |
| | | |
| Announce Request | 02h | 00h |
| Announce Response | 02h | 01h |
| Get New Lease Request | 02h | 02h |
| Get New Lease Response | 02h | 03h |
| Get Status Request | 02h | 04h |
| Get Status Response | 02h | 05h |
| Set RPL Location Request | 02h | 06h |
| Set RPL Location Response | 02h | 07h |
| Set Output Mode Request | 02h | 08h |
| Set Output Mode Response | 02h | 09h |
| Update Timeline Request | 02h | 0Ah |
| Update Timeline Response | 02h | 0Bh |
| Terminate Lease Request | 02h | 0Ch |
| Terminate Lease Response | 02h | 0Dh |
| Get Log Event List Request | 02h | 10h |
| Get Log Event List Response | 02h | 11h |
| Get Log Event Request | 02h | 12h |
| Get Log Event Response | 02h | 13h |

# Annex B Protocol Usage (Informative)

This section provides description of the how the protocol is used to ensure consistent behavior across products that implement the specification.

## B.1 Example Transaction