

SMPTE STANDARD

SMS-OMB Communications Protocol Specification



Table of Contents		Page
Foreword	3
Intellectual Property	3
1 Scope	3
2 Conformance Notation	3
3 Normative References	4
4 Terms and Definitions	4
4.1 Authority ID AuthId	4
4.2 Composition Playlist CPL	5
4.3 DataValue	5
4.4 DCI DCSS	5
4.5 Equipment Suite ES	5
4.6 Key Delivery Message KDM	5
4.7 IMB	5
4.8 Int32	5
4.9 Int64	5
4.10 Intra-Theater Messages ITM	5
4.11 Key Length Value KLV	5
4.12 Media-Block MB	5
4.13 OMB	5
4.14 Request-Response Pair RRP	5
4.15 Screen Management System SMS	6
4.16 Secure Processing Block SPB	6
4.17 Security Entity SE	6
4.18 Security Manager SM	6
4.19 Timestamp	6
4.20 Transport Layer Security TLS	6
4.21 UInt8	6
4.22 UInt32	6
4.23 Universal Label UL	6
4.24 Universally Unique Identifier UUID	6
5 Overview (Informative)	6
6 Intra-Theater Messages Protocol	7

SMPTE ST 430-17:2022

6.1	General requirements	7
6.2	TLS requirements	8
6.3	KLV requirements	8
6.4	Batch Data Types	8
7	API Commands.....	9
7.1	Request ID.....	9
7.2	Status Response	9
7.3	BadRequestResponse.....	9
7.4	getCertificate.....	10
7.5	adjustSecureTime.....	10
7.6	getSecureTime	11
7.7	validateCpl.....	11
7.8	getCplValidationResult	12
7.9	validateKdm	14
7.10	getKdmValidationResult	16
7.11	validatePlayback.....	17
7.12	invalidatePlayback.....	18
7.13	getDeviceInfo.....	18
7.14	getStatus.....	19
7.15	generateSecurityLogReport.....	21
7.16	getSecurityLogReport.....	21
7.17	cancelSecurityLogReport	22
7.18	generateCplProcessedLogReport	23
7.19	getCplProcessedLogReport	24
7.20	cancelCplProcessedLogReport.....	25
Annex A	(Normative) SMS-OMB Communications Protocol Defined-Length Universal Label (UL) Key.....	26
Annex B	(Normative) SMS-OMB Communications Protocol Group and Element Definitions.....	29
B.1	SMS-OMB Communications Protocol Group Node Definition.....	29
B.2	SMS-OMB Communications Protocol Element Node Definition	29
B.3	Fixed Array Type Node Definition.....	29
	Bibliography (Informative).....	30

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 its Standards Operations Manual. This SMPTE Engineering Document was prepared by Technology Committee 21DC Digital Cinema.

Intellectual Property

At the time of publication, no notice had been received by SMPTE claiming patent rights essential to the implementation of this Engineering Document. 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 describes an interface based on the KLV protocol for communications between an SMS client and an OMB server. It facilitates pre-playback validation of AuxData (SMPTE ST 429-14) and Immersive Audio Track Files (SMPTE ST 429-18) and the exchange of status information between an SMS and an OMB during playback.

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.

SMPTE ST 430-17:2022

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; then formal languages; then figures; and then any other language forms.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this engineering document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this engineering document are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE, ST 429-7:2006, "D-Cinema Packaging — Composition Playlist"

SMPTE, ST 429-14:2014, "D-Cinema Packaging — Aux Data Track File"

SMPTE, ST 429-18:2019, "D-Cinema Packaging — Immersive Audio Track File"

SMPTE, ST 430-1:2017, "D-Cinema Operations — Key Delivery Message"

SMPTE, ST 430-1:2017 Am1:2019, "D-Cinema Operations — Key Delivery Message"

SMPTE, ST 430-2:2017, "D-Cinema Operations — Digital Certificate"

SMPTE, ST 430-5:2011, "D-Cinema Operations — Security Log Event Class and Constraints"

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

SMPTE ST 2029:2009, "Uniform Resource Names for SMPTE Resources"

Internet Engineering Task Force (IETF), RFC 2246, "The TLS Protocol Version 1.0"¹

Internet Engineering Task Force (IETF), RFC 3268, "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)"¹

Internet Engineering Task Force (IETF), RFC 4122, "A Universally Unique Identifier (UUID) URN Namespace", 2005-07

Internet Engineering Task Force (IETF), RFC 5246, "The Transport Layer Security (TLS) Protocol Version 1.2"

Internet Engineering Task Force (IETF), RFC 8446, "The Transport Layer Security (TLS) Protocol Version 1.3"

4 Terms and Definitions

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

4.1 Authority ID

AuthId

as specified in SMPTE ST 430-5, that contains the identifier of the entity that authorized the action

¹ RFC 2246 and RFC 3268 are superseded by newer versions but are still referenced because support for TLS version 1.0 is required in section 6.2.

4.2 Composition Playlist**CPL**

as specified in SMPTE ST 429-7

4.3 DataValue

ordered sequence of data or metadata elements; the type of the data or metadata elements and the length of the ordered sequence are defined elsewhere

4.4 DCI DCSS

Digital Cinema Initiatives, Digital Cinema System Specification

Note: See Bibliography for more info

4.5 Equipment Suite**ES**

set of one or more *Secure Processing Blocks* (4.16) associated with one *Security Manager* (4.18) that collectively support essence playback through a projector

4.6 Key Delivery Message**KDM**

as specified in SMPTE ST 430-1

4.7 IMB

image *Media-Block* (4.12)

4.8 Int32

32-bit signed integer

4.9 Int64

64-bit signed integer

4.10 Intra-Theater Messages**ITM**

communications between the *Screen Management System* (4.15) and *Security Manager(s)* (4.18), and between the *Security Manager(s)* (4.18) and remote *Secure Processing Blocks* (4.16)

4.11 Key Length Value**KLV**

as defined in SMPTE ST 336

4.12 Media-Block**MB**

Secure Processing Block (4.16) that contains a *Security Manager* (4.18) and performs media decryption

4.13 OMB

outboard *Media-Block* (4.12)

4.14 Request-Response Pair**RRP**

single interaction between the *Security Entities* (4.17)

SMPTE ST 430-17:2022

4.15 Screen Management System

SMS

(non-secure) *Security Entity* (4.17) that directs security functions for a single auditorium on behalf of exhibition management

4.16 Secure Processing Block

SPB

Security Entity (4.17) which provides a physical and logical protection perimeter around other *Security Entities* (4.17)

4.17 Security Entity

SE

logical processing device which executes a distinct security process or function

4.18 Security Manager

SM

Security Entity (4.17) that is implemented within a *Media-Block* (4.12), responsible for parsing *Key Delivery Messages* (4.6) and generating log records

Note: There can be more than one *Media-Block* (4.12), and therefore more than one *Security Manager*, associated with an auditorium within an exhibition site.

4.19 Timestamp

time representing the number of seconds elapsed since January 1, 1970 00:00:00 UTC

4.20 Transport Layer Security

TLS

as defined in IETF RFC 2246

4.21 UInt8

8-bit unsigned integer

4.22 UInt32

32-bit unsigned integer

4.23 Universal Label

UL

as defined in SMPTE ST 2029

4.24 Universally Unique Identifier

UUID

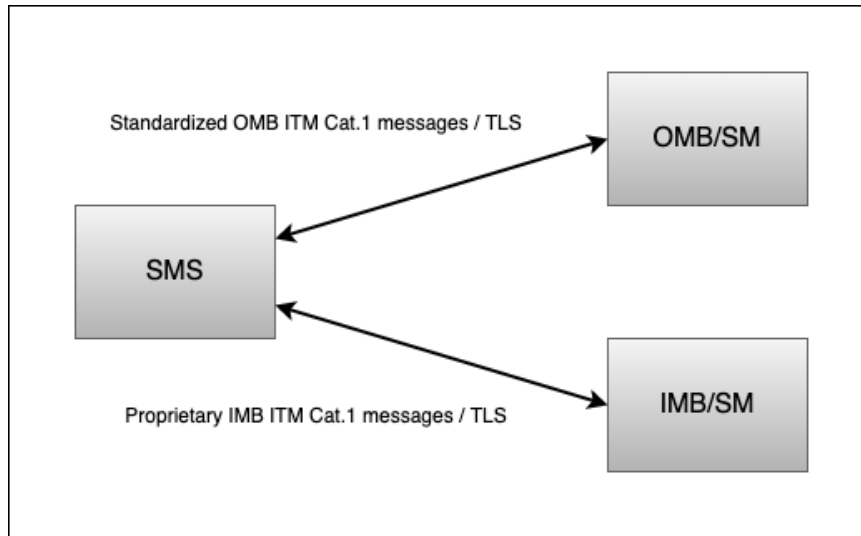
as specified in IETF RFC 4122

5 Overview (Informative)

The protocol described in this document is designed to facilitate interoperability between Digital Cinema equipment in an auditorium. Specifically, it is designed to allow SMS entities to operate with the security subsystem (SM) of external media-blocks (OMBs) the same way it needs to operate with the SM of the image media-block (IMB), as shown in Figure 1 below.

While the protocol defined herein is directed to SMS-to-OMB communication functions, in recognition of the similarity to SMS-to-IMB communication functions, implementers are encouraged to consider adopting this protocol and commands where beneficial for SMS-to-IMB communication functions.

Figure 1 – SMS Message Exchange with Media Blocks



The protocol has been designed to meet the non-normative requirements of DCI DCSS section 9.4.5.2.4 for Category 1 operational messages. Supported commands are:

- CPL and KDM authentication;
- Playback conditions validation;
- Secure time adjustment;
- Security log report export; and
- CPLProcessed log report export.

Such functionalities allow an SMS to initiate security functions as a pre-condition to authorize the playback and ensure valid payout.

The request/response commands herein are expected to draw information from validation functions already required of Media-Block (MB) Security Managers (SM). Therefore, in implementing compliance to these commands, MB designs can consider avoidance of duplication of processing, storage and/or logging records beyond those already required.

6 Intra-Theater Messages Protocol

6.1 General requirements

The protocol is designed to carry all commands using Request-Response Pair (RRP) messages. Therefore, each command can be defined as an RRP.

- The SMS shall be the Initiator, i.e., it initiates the connection and sends Request messages.
- The OMB SM shall be the Responder, i.e., it responds to an RRP request with the matching RRP Response message.
- The protocol is synchronous:

SMPTE ST 430-17:2022

- The Initiator shall not initiate a new RRP until the response to the previous one has been received.
- Long processes are started with an RRP and then polled with additional RRP's to determine when the process is complete. RRP's shall be designed to complete in a timely manner, i.e., avoid blocking commands.
- 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, implementations shall have a maximum round-trip Request-to-Response latency of 5 seconds. Latency shall be measured from the end of the "Request" message receipt to the start of the "Response" message transmission.
- If the OMB is too busy to accept or process a request, it shall send the corresponding response with the Status Response set to ResponderBusy.
- If the Status Response is not set to "RRP successful", all other items in the Pack are irrelevant but shall be present as per SMPTE ST 336.
- Should the OMB fail to respond (at all) after the 5 second time limit, the SMS may consider this a communications failure condition and may close the TCP connection.
- RRP messages are encoded using SMPTE ST 336 Key-Length-Value (KLV) triplets and Defined-Length Pack encoding. The Pack Key ULs are defined in Table A.1 in Annex A. The node definitions containing the protocol's Packs and Elements are defined in Annex B.
- Protocol security is met using TLS tunneling.

6.2 TLS requirements

- TLS encryption shall be used as described in this section. TLS version 1.3 as defined in IETF RFC 8446 should be used. For purposes of interoperability, TLS version 1.0 as defined in IETF RFC 2246 shall be supported. TLS version 1.2 as defined in IETF RFC 5246 may be supported.
- The cipher suite shall be "TLS_RSA_WITH_AES_128_CBC_SHA" as defined in IETF RFC 3268 when TLS version 1.0 is used.

Those requirements ensure usage of strong cryptography and well-known algorithms amongst Digital Cinema equipment.

- Digital certificates exchanged during the TLS handshake shall be compliant with SMPTE ST 430-2.
- Server and client certificate exchange shall be required but peer authentication is not mandated. Specifically, certificate chain validation context shall not require the peer root certificate to be trusted (see SMPTE ST 430-2 section 6.2 bullet 19).
- IP addresses and port numbers for the TLS connection shall be configurable by the end user.

Those requirements ensure device identity can be exchanged for the purpose of out-of-band operations (e.g. logging or UI information).

6.3 KLV requirements

- KLV Length fields shall be encoded in BER long-form with a fixed length of four bytes (Defined-Length Pack).
- Multi-byte integer values shall be encoded using Big Endian byte-ordering.

6.4 Batch Data Types

Messages defined in this document may contain batches. A batch is a compound data type that is created from combinations of simple data types. It is usually preceded by a name (e.g., a CplValidationBatch is an unordered batch of CPL validation data).

Batch: A compound type comprising multiple individual elements. The elements are unordered, the type is defined, the count of elements is explicit and the size of each element is fixed and explicit.

xxxBatch: A batch of zero or more individual elements of name “xxx” preceded by a header of 8 bytes, as defined in Table 1 below. The first 4 bytes of the header define the number of elements to follow and the second 4 bytes define the length of each element, both represented as UInt32. In case Number of Elements field is 0, the batch is empty, the Element Length field is irrelevant, and the batch does not contain any elements after the header bytes.

Table 1 – Item Definitions for Type: xxxBatch

Item Name	Type	Len	Meaning
Number of Elements	UInt32	4	The number of Elements in the Batch
Element Length	UInt32	4	The length of each Element
First component of first instance of xxx	First of zero or more components describing element ‘xxx’ and having a total length of Element Length

7 API Commands

7.1 Request ID

RRP Request and Response message shall contain a Request ID element. It shall be a 32-bit unsigned integer uniquely identifying the RRP.

It is the duty of the Initiator (SMS) to ensure the Request ID will be unique within a single TLS session. A basic scheme could be to simply increase the Request ID by one for each new RRP.

A Responder shall copy the Request ID from the RRP Request into the RRP Response.

7.2 Status Response

Each RRP Response message shall contain a Status Response, as defined in Table 2 below.

Table 2 – Status Response

Item Name	Value	Meaning
RRP successful	0	Request successfully processed
RRP failed	1	Request failed to be successfully processed
RRP invalid	2	Invalid parameter or command structure
ResponderBusy	3	Responder is too busy to process the Request
-	4-255	Reserved

7.3 BadRequestResponse

The purpose of this message is to handle unknown RRP Requests, i.e., a request for which the KLV Key is invalid. It defines a generic Response message.

A Responder shall reply with a BadRequestResponse, as defined in Table 3 below, and its Status Response set to “RRP invalid” if an unknown Request is received.

Table 3 – OMBBadRequestResponse Pack

Item Name	Type	Len	Meaning
OMBBadRequestResponse	Pack Key	16	Bad Request Response
Length	BER Length	4	Pack length
Status Response	UInt8	1	General response info

7.4 getCertificate

This command allows retrieving SM and Log Signer certificates from the OMB. The request for this command is defined in Table 4 below.

Table 4 – GetCertificateRequest Pack

Item Name	Type	Len	Meaning
GetCertificateRequest UL	Pack Key	16	getCertificate Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request
Certificate Kind	UInt8	1	Certificate Kind (see Table 5)

Table 5 - Certificate Kind

Enumerate	Value	Description
CERTIFICATE_KIND_SM	0	OMB SM device certificate
CERTIFICATE_KIND_LS	1	OMB Log Signer certificate
	2-255	Reserved

The response for this command, as defined in Table 6 below, shall contain the digital certificate of the OMB, encoded in DER ASN.1 form as per SMPTE ST 430-2.

Table 6 - GetCertificateResponse Pack

Item Name	Type	Len	Meaning
GetCertificateResponse UL	Pack Key	16	getCertificate Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
getCertificate Status	UInt8	1	Error Code (see Table 7)
Certificate Length	UInt32	4	Certificate data length
Certificate Data	DataValue	Var	DER encoded certificate data

Table 7 - getCertificate Status Error Codes (not used yet)

Enumerate	Value	Description
NO_ERROR	0	No Error
-	1-255	Reserved

7.5 adjustSecureTime

This command allows adjusting the OMB secure clock within the range specified by the DCI DCSS (+6,-6 minutes per calendar year). The request and response for this command are defined in Table 8 and Table 9 below, respectively.

If the adjustment exceeds this range the response shall contain the Adjustment Status SECURITY_CLOCK_ADJUST_OUT_OF_RANGE.

Table 8 - AdjustSecureTimeRequest Pack

Item Name	Type	Len	Meaning
AdjustSecureTimeRequest UL	Pack Key	16	adjustSecureTime Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request
Offset	Int32	4	Offset value in seconds
AuthId Text Length	UInt32	4	AuthId text length in bytes
AuthId Text	DataValue	Var	AuthId text

Table 9 - AdjustSecureTimeResponse Pack

Item Name	Type	Len	Meaning
AdjustSecureTimeResponse UL	Pack Key	16	adjustSecureTime Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
adjustSecureTime Status	UInt8	1	Error Code (see Table 10)

Table 10 - adjustSecureTime Status Error Codes

Enumerate	Value	Description
NO_ERROR	0	No Error
SECURITY_CLOCK_ADJUST_OUT_OF_RANGE	1	Clock adjustment out of allowed range - adjustment rejected and no clock adjustment made
SECURITY_CLOCK_ADJUST_FAILED	2	Clock adjustment failed - adjustment not possible
-	3-255	Reserved

7.6 getSecureTime

This command is used to retrieve the current OMB secure clock time. The request and response for this command are defined in Table 11 and Table 12 below, respectively.

Table 11 - GetSecureTimeRequest Pack

Item Name	Type	Len	Meaning
GetSecureTimeRequest UL	Pack Key	16	getSecureTime request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request

Table 12 - GetSecureTimeResponse Pack

Item Name	Type	Len	Meaning
GetSecureTimeResponse UL	Pack Key	16	getSecureTime Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
Secure Clock Time	Int64	8	Timestamp representing the OMB secure clock time
getSecureTime Status	UInt8	1	Error Code (see Table 13)

Table 13 - getSecureTime Status Error Codes

Enumerate	Value	Description
NO_ERROR	0	No Error
SECURITY_CLOCK_RETRIEVAL_FAILED	1	Clock retrieval failed
-	2-255	Reserved

7.7 validateCpl

This command allows submitting a CPL to the OMB SM for validation purposes. The request and response for this command are defined in Table 14 and Table 16 below, respectively.

The OMB SM shall validate whether the CPL is in compliance with SMPTE ST 429-7 and the CPLCheck Record of SMPTE ST 430-5. The validation process shall be asynchronous: The Initiator shall poll for the result using the getCplValidationResult command.

In order to fulfill playback verification duties, the OMB should retain CPL validation data until the SMS sends the next invalidatePlayback command.

SMPTE ST 430-17:2022

For each CPL, the SMS shall provide a list of asset UUIDs and computed hash values for each asset contained in the CPL to the OMB in order for the OMB to perform its security log duties as defined in SMPTE ST 430-5.

Note: The protocol allows the requester to start several validations in parallel, but the OMB SM implementation might have some limitations. It is allowed to reject a new validation until some pending validations are completed by replying with a “ResponderBusy” status.

Table 14 - ValidateCPLRequest Pack

Item Name	Type	Len	Meaning
ValidateCPLRequest UL	Pack Key	16	validateCpl Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request
AuthId Text Length	UInt32	4	AuthId text length in bytes
AuthId Text	DataValue	Var	AuthId text
CPL XML Data Length	UInt32	4	CPL data length
CPL XML Data	DataValue	Var	CPL data
AssetIntegrityHash Batch	AssetIntegrityHashBatch	8 + 36n	An unordered batch of Asset ID and associated Asset Hash values (see Table 15)

Table 15 - AssetIntegrityHashBatch

Item Name	Type	Len	Meaning
Number of Items	UInt32	4	The number of batch items
Item Length	UInt32	4	The length of the batch item (36 bytes)
Asset ID	UInt8Array16	16	Asset UUID
Asset Hash	UInt8Array20	20	Hash of the asset

Table 16 - ValidateCPLResponse Pack

Item Name	Type	Len	Meaning
ValidateCPLResponse UL	Pack Key	16	validateCpl Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
validateCpl Status	UInt8	1	Error Code (see Table 17)

Table 17 - validateCpl Status Error Codes

Enumerate	Value	Description
NO_ERROR	0	No Error
-	1-255	Reserved

7.8 getCplValidationResult

This command allows polling the validation status for a specific CPL. The request and response for this command are defined in

Table 18 and Table 19 below, respectively.

The OMB shall return the pending status if the CPL validation is still ongoing. The getCplValidationResult Status is encoded as a bit field that can contain multiple indicated errors.

Table 18 - GetCPLValidationResultRequest Pack

Item Name	Type	Len	Meaning
GetCPLValidationResultRequest UL	Pack Key	16	getCplValidationResult Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request
CPL ID	UInt8Array16	16	CPL UUID

Table 19 - GetCPLValidationResultResponse Pack

Item Name	Type	Len	Meaning
GetCPLValidationResultResponse UL	Pack Key	16	getCplValidationResult Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
getCplValidationResult Status	UInt8	1	Error Code (see Table 20)

Table 20 - getCplValidationResult Status Error Codes

Enumerate	Value	Description
NO_ERROR	0	No Error
CPL_VALIDATION_PENDING	1	CPL validation is ongoing (not an error)
CPL_NOT_COMPLIANT_SMPTE_429_7	2	CPL does not meet normative language of SMPTE ST S429-7
CPL_INVALID_DSIG	4	Invalid CPL signature
CPL_INVALID_DSIG_CHAIN	8	Invalid CPL signer chain
CERT_FORMAT_ERROR	16	Signer Certificate does not meet normative language of SMPTE ST 430-2
ASSET_HASH_ERROR	32	At least one computed asset hash provided by the SMS does not match the CPL asset hash
ASSET_MISSING_ERROR	64	At least one asset is included in the CPL which does not have a matching batch in the AssetIntegrityHashBatch
-	128	Reserved

7.9 validateKdm

This command allows submitting a KDM to an OMB SM for validation purposes. The request and response for this command are defined in

Table 21 and Table 22 below, respectively.

The OMB SM shall validate whether the KDM is in compliance with SMPTE ST 430-1. The validation process shall be asynchronous: The Initiator shall poll for the result using the `getKdmValidationResult` command.

In order to fulfill playback verification duties, the OMB should retain KDM validation data until the SMS sends the next `invalidatePlayback` command.

The KDM validation shall fail if the target CPL is unknown, i.e., it has not been previously validated.

Note 1: The protocol allows the requester to start several validations in parallel, but the OMB SM implementation might have some limitations. It is allowed to reject a new validation until some pending validations are completed by replying with a “ResponderBusy” status.

Note 2: KDM validation by the OMB also requires decryption, which is a time consuming process. Typically, the SMS will request CPL and KDM validation when the content is selected in the Player, i.e., ahead of actual playback.

Table 21 - ValidateKDMRequest Pack

Item Name	Type	Len	Meaning
ValidateKDMRequest UL	Pack Key	16	validateKdm Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request
AuthId Text Length	UInt32	4	AuthId text length in bytes
AuthId Text	DataValue	Var	AuthId text
KDM XML Data Length	UInt32	4	KDM XML data length
KDM XML Data	DataValue	Var	KDM XML data

Table 22 - ValidateKDMResponse Pack

Item Name	Type	Len	Meaning
ValidateKDMResponse UL	Pack Key	16	validateKdm Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
validateKdm Status	UInt8	1	Error Code (see Table 23)

Table 23 - validateKdm Status Error Codes

Enumerate	Value	Description
NO_ERROR	0	No Error
-	1-255	Reserved

7.10 getKdmValidationResult

This command allows polling the validation status for a specific KDM. The request and response for this command are defined in Table 24 and Table 25 below, respectively.

The OMB shall return the pending status if the KDM validation is still ongoing. The getKdmValidationResult Status is encoded as a bit field that can contain multiple indicated errors.

Table 24 - GetKDMValidationResultRequest Pack

Item Name	Type	Len	Meaning
GetKDMValidationResultRequest UL	Pack Key	16	getKdmValidationResult Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request
KDM ID	UInt8Array16	16	ID of the KDM

Table 25 - GetKDMValidationResultResponse Pack

Item Name	Type	Len	Meaning
GetKDMValidationResultResponse UL	Pack Key	16	getKdmValidationResult Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
getKdmValidationResult Status	UInt32	4	Error Code (see Table 26)

Table 26 - getKdmValidationResult Status Error Codes

Enumerate	Value	Description
NO_ERROR	0	No Error
KDM_VALIDATION_PENDING	1	KDM validation is ongoing
KDM_NOT_COMPLIANT SMPTE 430_1	2	KDM is not compliant with SMPTE ST 430-1
KDM_INVALID_DSIG_CHAIN	4	Invalid KDM signer chain
KDM_INVALID_DSIG_DGST	8	Invalid KDM signature digest
KDM_INVALID_DSIG	16	Invalid KDM signature

KDM_INVALID_TDL_DEVICE	32	Invalid TDL device
KDM_INVALID_TDL	64	Invalid TDL
KDM_BAD_RECIPIENT	128	KDM is not targeting the OMB
KDM_DECRYPTION_FAILED	256	KDM decryption error
KDM_NOT_SUBMITTED	512	KDM ID was not submitted in any pending validateKdm Request
-	All other bit values	Reserved

7.11 validatePlayback

This command requests the OMB to validate the playback of the track files that the OMB has been designated to process for a given playlist which is represented by a list of CPLs. The request and response for this command are defined in Table 27 and Table 28 below, respectively.

The SMS is provided with the following information by the OMB:

- Whether playback is permitted or not at the time of the request.
- The time at which the OMB will revoke playback permission.

A CplValidationBatch “Error Code” item not equal to “NO_ERROR” is an indication that the playback of this CPL is not permitted by the SM of the OMB.

The “CPLProcessed Report Delinquency Time” item shall be the timestamp when the CPLProcessed event collection will become delinquent and playback permissions will be revoked.

The “Permitted Playback Start Time” item shall report the value of the “ContentKeysNotValidBefore” field of the KDM selected for playback. In case unencrypted content is validated or the KDM for the content is not available the “Permitted Playback Start Time” shall be set to 0.

The “Expiration Time” item shall be the timestamp when the playback permissions of the corresponding CPL will be revoked by the SM of the OMB. In case unencrypted content is validated the “Expiration Time” shall be set to 0x7FFFFFFFFFFFFFFF (max. value of Int64 type). In case the KDM for the content is not available the “Expiration Time” shall be set to 0.

If there are no uncollected CPLStart or CPLStop events available, the CPLProcessed Report Delinquency Time shall be set to 0x7FFFFFFFFFFFFFFF.

Table 27 - ValidatePlaybackRequest Pack

Item Name	Type	Len	Meaning
ValidatePlaybackRequest UL	Pack Key	16	validatePlayback Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request
AuthId Text Length	UInt32	4	AuthId text length in bytes
AuthId Text	DataValue	Var	AuthId text

Table 28 - ValidatePlaybackResponse Pack

Item Name	Type	Len	Meaning
ValidatePlaybackResponse UL	Pack Key	16	validatePlayback Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
CplValidation Batch	CplValidationBatch	8 + 36n	An unordered batch of CPL validation results (see Table 29)
CPLProcessed Report Delinquency Time	Int64	8	Timestamp when the CPLProcessed event collection by the SMS becomes delinquent

Table 29 - CplValidationBatch

Item Name	Type	Len	Meaning
Number of Items	UInt32	4	The number of batch items
Item Length	UInt32	4	The length of the batch item
CPL ID	UInt8Array16	16	CPL UUID
Error Code	UInt32	4	Error Code (see Table 30)
Permitted Playback Start Time	Int64	8	Timestamp when CPL becomes playable
Expiration Time	Int64	8	Timestamp when CPL is no longer playable

Table 30 - Error Codes

Enumerate	Value	Description
NO_ERROR	0	No Error
CPL_KDM_MISSING	1	Missing KDM
CPL_KDM_NOT_YET_VALID	2	KDM is not yet valid
CPL_KDM_EXPIRED	4	KDM is expired
CPL_INVALID_CONTENT_AUTHENTICATOR	8	Invalid ContentAuthenticator
KDM_INVALID_KEY_MATCH	16	KDM keys not matching the CPL
CPLPROCESSED_DELINQUENT	32	Delinquent CPLProcessed records exist
-	All other bit values	Reserved

7.12 invalidatePlayback

This command tells the OMB SM to purge all the keys and validation data. This is typically done when the content is removed from the player. The request and response for this command are defined in Table 31 and Table 32 below, respectively.

Table 31 - InvalidatePlaybackRequest Pack

Item Name	Type	Len	Meaning
InvalidatePlaybackRequest UL	Pack Key	16	invalidatePlayback Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response

Table 32 - InvalidatePlaybackResponse Pack

Item Name	Type	Len	Meaning
InvalidatePlaybackResponse UL	Pack Key	16	invalidatePlayback Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info

7.13 getDeviceInfo

The command allows retrieving some information from the OMB Security Manager. The request and response for this command are defined in Table 33 and Table 34 below, respectively.

Table 33 - GetDeviceInfoRequest Pack

Item Name	Type	Len	Meaning
GetDeviceInfoRequest UL	Pack Key	16	Identifies getDeviceInfo Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request

Table 34 - GetDeviceInfoResponse Pack

Item Name	Type	Len	Meaning
GetDeviceInfoResponse UL	Pack Key	16	Identifies getDeviceInfo Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
Identifier Text Length	UInt32	4	The length of the identifier text
Identifier Text	DataValue	Var	Product Name
Version Text Length	UInt32	4	The length of the version text
Version Text	DataValue	Var	The software version text
Supported DataEssenceCoding ULs	DataEssenceCodingBatch	8 + 16n	Batch of DataEssenceCoding ULs supported for playback by this device (see Table 35)

Table 35 - DataEssenceCodingBatch

Item Name	Type	Len	Meaning
Number of Items	UInt32	4	The number of batch items
Item Length	UInt32	4	The length of the batch item (16 bytes)
DataEssenceCoding UL	UInt8Array16	16	UL defining an AuxData track file DataEssenceCoding ID that is supported by this device

7.14 getStatus

This command retrieves the status. It can be used as a heartbeat function to maintain the connection. The request and response for this command are defined in Table 36 and Table 37 below, respectively.

If there are no uncollected CPLStart or CPLStop events available, the CPLProcessed Report Delinquency Time shall be set to 0x7FFFFFFFFFFFFFFF (max. value of Int64 type).

Table 36 - GetStatusRequest Pack

Item Name	Type	Len	Meaning
GetStatusRequest UL	Pack Key	16	getStatus Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Protocol Version	UInt8	1	Protocol version used by the client, shall be set to 0

Table 37 - GetStatusResponse Pack

Item Name	Type	Len	Meaning
GetStatusResponse UL	Pack Key	16	getStatus Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
Protocol Version	UInt8	1	Protocol version used by the server, shall be set to 0
OMB Status	UInt8	1	OMB Status (see Table 38)
CPLProcessed Report Delinquency Time	Int64	8	Timestamp when CPLProcessed event collection by the SMS becomes delinquent

Table 38 - OMB Status

Enumerate	Value	Description
NO_SHOW_LOADED	0	No show is loaded
SHOW_LOADED_STOPPED	1	Show is loaded and device is in stopped mode
DEVICE_PAUSED	2	Device is in paused mode
DEVICE_PLAYING	3	Device is playing
DEVICE_IN_ERROR	4	Device is in error state
PROTOCOL_NOT_SUPPORTED	5	Protocol version provided in getStatus Request not supported by the OMB
-	6-255	Reserved

7.15 generateSecurityLogReport

This command triggers the generation of a signed XML security log report as defined in SMPTE ST 430-5. The request and response for this command are defined in Table 39 and Table 40 below, respectively.

The log generation process shall be asynchronous: The Initiator shall poll for the result using the getSecurityLogReport command.

Table 39 - GenerateSecurityLogReportRequest Pack

Item Name	Type	Len	Meaning
GenerateSecurityLogReportRequest UL	Pack Key	16	generateSecurityLogReport Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request
Start Time	Int64	8	Timestamp representing the beginning of the time window containing the requested security log events
End Time	Int64	8	Timestamp representing the end of the time window containing the requested security log events

Table 40 - GenerateSecurityLogReportResponse Pack

Item Name	Type	Len	Meaning
GenerateSecurityLogReportResponse UL	Pack Key	16	generateSecurityLogReport Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
Generation Status	UInt8	1	Generation Status (see Table 41)

Table 41 - Generation Status

Enumerate	Value	Description
NO_ERROR	0	Generation request has been accepted.
GENERATION_ALREADY_STARTED	1	Generation request is rejected because a log generation request is already in progress.
GENERATION_CANCELED	2	Generation request is rejected because a log cancellation request is in progress.
-	3-255	Reserved

7.16 getSecurityLogReport

This command triggers the export of the XML security log report data. The request and response for this command are defined in Table 42 and Table 43 below, respectively.

Because no assumption can be made about the OMB SM resources (e.g. RAM or internal storage), the OMB SM has the opportunity to report smaller partial reports.

SMPTE ST 430-17:2022

An Export Status of REPORT_IS_PARTIAL indicates to the requester that it must complete its original request by further extracting another report by using the Next Report Start Time timestamp in the subsequent request. If the value of Export Status is not REPORT_IS_PARTIAL, the value of Next Report Start Time shall be set to 0.

If there are no Log Records available in the specified time window, an Export Status value of NO_ERROR and an XML Report Data Length of zero shall be contained in the response.

If the value of Export Status is any value except NO_ERROR or REPORT_IS_PARTIAL, the value of XML Report Data Length shall be set to 0.

Table 42 - GetSecurityLogReportRequest Pack

Item Name	Type	Len	Meaning
GetSecurityLogReportRequest UL	Pack Key	16	getSecurityLogReport Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request

Table 43 - GetSecurityLogReportResponse Pack

Item Name	Type	Len	Meaning
GetSecurityLogReportResponse UL	Pack Key	16	getSecurityLogReport Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
Export Status	UInt8	1	Export Status (see Table 44)
Next Report Start Time	Int64	8	Timestamp representing the beginning of the time window for the next request
XML Report Data Length	UInt32	4	XML report data length
XML Report Data	DataValue	Var	XML report data

Table 44 - Export Status

Enumerate	Value	Description
NO_ERROR	0	The XML report is available in the returned DataValue.
REPORT_IS_PARTIAL	1	The XML report is available in the returned DataValue but it does not cover the whole requested time window.
GENERATION_IN_PROGRESS	2	The XML report is still being generated. The requester shall retry later.
GENERATION_CANCELED	3	The XML report generation has been cancelled.
GENERATION_ERROR	4	The XML report generation failed.
NO_PENDING_REQUEST	5	There is no generateSecurityLogReport request that hasn't been fulfilled.
-	6-255	Reserved

7.17 cancelSecurityLogReport

This command cancels any currently open security log report generation request. The request and response for this command are defined in Table 45 and Table 46 below, respectively.

Since cancellation may not be an immediate operation, the OMB SM:

- Shall respond to further generateSecurityLogReport requests with the Generation Status set to GENERATION_CANCELED until ready to proceed.
- Shall respond to further getSecurityLogReport requests with the Export Status set to GENERATION_CANCELED until ready to proceed.

Table 45 - CancelSecurityLogReportRequest Pack

Item Name	Type	Len	Meaning
CancelSecurityLogReportRequest UL	Pack Key	16	cancelSecurityLogReport Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request

Table 46 - CancelSecurityLogReportResponse Pack

Item Name	Type	Len	Meaning
CancelSecurityLogReportResponse UL	Pack Key	16	cancelSecurityLogReport Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info

7.18 generateCplProcessedLogReport

This command triggers the generation of a signed XML security log report as defined in SMPTE ST 430-5. The request and response for this command are defined in Table 47 and Table 48 below, respectively.

The log generation process shall be asynchronous: The Initiator shall poll for the result using the getCplProcessedLogReport command.

In response to receipt of this command, the OMB SM is expected to generate a log report consisting only of previously unreported CPLStart and CPEnd security log records as defined in SMPTE ST 430-5.

Table 47 - GenerateCPLProcessedLogReportRequest Pack

Item Name	Type	Len	Meaning
GenerateCPLProcessedLogReportRequest UL	Pack Key	16	generateCplProcessedLogReport Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request

Table 48 - GenerateCPLProcessedLogReportResponse Pack

Item Name	Type	Len	Meaning
GenerateCPLProcessedLogReportResponse UL	Pack Key	16	generateCplProcessedLogReport Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
Generation Status	UInt8	1	Generation Status (see Table 49)

Table 49 - Generation Status

Enumerate	Value	Description
NO_ERROR	0	Generation request has been accepted.
GENERATION_ALREADY_STARTED	1	Generation request is rejected because a log generation request is already in progress.
GENERATION_CANCELED	2	Generation request is rejected because a log cancellation request is in progress.
-	3-255	Reserved

7.19 getCplProcessedLogReport

This command triggers the export of the requested XML CPLProcessed security log report which consists of previously unreported CPLStart and CPEnd security log records as defined in section 7.18. The request and response for this command are defined in Table 50 and Table 51 below, respectively.

An Export Status of REPORT_IS_PARTIAL indicates to the requester that the report does not contain all unreported CPLProcessed event records, and it should issue another generateCplProcessedLogReport request.

If there are no previously unreported CPLStart or CPEnd Log Records to report, an Export Status value of NO_ERROR and an XML Report Data Length of zero shall be contained in the response.

Table 50 - GetCPLProcessedLogReportRequest Pack

Item Name	Type	Len	Meaning
GetCPLProcessedLogReportRequest UL	Pack Key	16	getCplProcessedLogReport Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request

Table 51 - GetCPLProcessedLogReportResponse Pack

Item Name	Type	Len	Meaning
GetCPLProcessedLogReportResponse UL	Pack Key	16	getCplProcessedLogReport Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info
Export Status	UInt8	1	Export Status (see Table 52)
XML Report Data Length	UInt32	4	XML report data length Must be 0 if the Export Status is not 0 or 1
XML Report Data	DataValue	Var	XML report data

Table 52 - Export Status

Enumerate	Value	Description
NO_ERROR	0	The XML log report is available in the returned DataValue.
REPORT_IS_PARTIAL	1	The XML log report is available in the returned DataValue but it does not include all unreported CPLProcessed event records. The requester should issue another generation request.
GENERATION_IN_PROGRESS	2	The XML log report is still being generated. The requester shall retry later.
GENERATION_CANCELED	3	Generation request is rejected because a cancelCplProcessedLogReport request is in progress.
GENERATION_ERROR	4	The XML report generation failed.
NO_PENDING_REQUEST	5	There is no getCplProcessedLogReport request that hasn't been fulfilled.
-	6-255	Reserved

7.20 cancelCplProcessedLogReport

This command cancels any currently open generateCplProcessedLogReport request. The request and response for this command are defined in Table 53 and Table 54 below, respectively.

Since cancellation may not be an immediate operation, the OMB SM:

- Shall respond to further generateCplProcessedReport requests with the Generation Status set to GENERATION_CANCELED until ready to proceed.
- Shall respond to further getCplProcessedReport requests with the Export Status set to GENERATION_CANCELED until ready to proceed.

Table 53 - CancelCPLProcessedLogReportRequest Pack

Item Name	Type	Len	Meaning
CancelCPLProcessedLogReportRequest UL	Pack Key	16	cancelCplProcessedLogReport Request
Length	BER Length	4	Pack length
Request ID	UInt32	4	ID of this Request

Table 54 - CancelCPLProcessedLogReportResponse Pack

Item Name	Type	Len	Meaning
CancelCPLProcessedLogReportResponse UL	Pack Key	16	cancelCplProcessedLogReport Response
Length	BER Length	4	Pack length
Request ID	UInt32	4	IDs the Request for which this is the Response
Status Response	UInt8	1	General Response info

Annex A (Normative) SMS-OMB Communications Protocol Defined-Length Universal Label (UL) Key

As a Defined-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 Values for all SMS-OMB Communications Protocol Pack Keys

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	Defined-Length Pack
7	Structure Designator	01h	Groups Dictionary
8	Version Number	01h	Registry Version: Dictionary version 1
9	Item Designator	0Ch	Compound
10	Group ID (1)	03h	Messages
11	Group ID (2)	02h	SMS-OMB-Communications Protocol
12	Set/Pack Kind	xx	Command Type (see Table A.2)
13	Reserved	00h	Not assigned
14	Reserved	00h	Not assigned
15	Reserved	00h	Not assigned
16	Reserved	00h	Not assigned

The values of byte 12 for specified SMS-OMB Communications Protocol Pack Keys are given in Table A.2 below.

Table A.2 - Set/Pack Kind Values for Pack Keys

SMPTE UL byte 12 node names	Byte 12
OMBBadRequestResponse	01h
GetCertificateRequest	02h
GetCertificateResponse	03h
AdjustSecureTimeRequest	04h
AdjustSecureTimeResponse	05h
GetSecureTimeRequest	06h
GetSecureTimeResponse	07h
ValidateCPLRequest	08h
ValidateCPLResponse	09h
GetCPLValidationResultRequest	0Ah
GetCPLValidationResultResponse	0Bh
ValidateKDMRequest	0Ch
ValidateKDMResponse	0Dh
GetKDMValidationResultRequest	0Eh
GetKDMValidationResultResponse	0Fh
ValidatePlaybackRequest	10h
ValidatePlaybackResponse	11h
InvalidatePlaybackRequest	12h
InvalidatePlaybackResponse	13h
GetDeviceInfoRequest	14h

GetDeviceInfoResponse	15h
GetStatusRequest	16h
GetStatusResponse	17h
GenerateSecurityLogReportRequest	18h
GenerateSecurityLogReportResponse	19h
GetSecurityLogReportRequest	1Ah
GetSecurityLogReportResponse	1Bh
CancelSecurityLogReportRequest	1Ch
CancelSecurityLogReportResponse	1Dh
GenerateCPLProcessedLogReportRequest	1Eh
GenerateCPLProcessedLogReportResponse	1Fh
GetCPLProcessedLogReportRequest	20h
GetCPLProcessedLogReportResponse	21h
CancelCPLProcessedLogReportRequest	22h
CancelCPLProcessedLogReportResponse	23h

Annex B (Normative) SMS-OMB Communications Protocol Group and Element Definitions

B.1 SMS-OMB Communications Protocol Group Node Definition

The SMS-OMB Communications Protocol group is a newly defined node-type group that is a child of the Messages group node and that provides for reasonable grouping of the RRP related Defined-Length Packs defined in Annex A. The SMS-OMB-Communications Protocol group shall be defined as specified in Table B.1 below.

Table B.1 - SMS-OMB Communications Protocol group

Kind	Node
Name	SMS-OMB Communications Protocol
Symbol	SMSOMBCommunicationsProtocol
UL	urn:smpte:ul:060e2b34.027f0101.0c030200.00000000
Definition	Protocol Metadata Packs for SMS-OMB Communication

Note: The value of byte 6 (0x7F) in the SMS-OMB Communications Protocol group is a placeholder that is used in the registers but is not used in actual implementations of the protocol, where Defined-Length Pack coding (value 0x05) is used.

B.2 SMS-OMB Communications Protocol Element Node Definition

The SMS-OMB Communications Protocol element is a newly defined node-type element that is a child of the Protocols element node and that provides for reasonable grouping of the protocol element items defined in this document. The SMS-OMB Communications Protocol element shall be defined as specified in Table B.2 below.

Table B.2 - SMS-OMB Communications Protocol element

Kind	Node
Name	SMS-OMB Communications Protocol
Symbol	SMSOMBCommunicationsProtocol
UL	urn:smpte:ul:060e2b34.0101010e.02400200.00000000
Definition	Elements defined as part of the SMS-OMB Communications protocol

B.3 Fixed Array Type Node Definition

The UInt8Array20 type is a newly defined type that is a child of the InseparableBytestrings type and that provides the type for 20-byte hash elements used in this document. The UInt8Array20 type shall be defined as specified in Table B.3 below.

Table B.3 - UInt8Array20 type

Kind	Node
Name	UInt8Array20
Symbol	UInt8Array20
UL	urn:smpte:ul: 060e2b34.01040101.01030400.00000000
Definition	Fixed-size array of 20 bytes

Bibliography (Informative)

Digital Cinema Initiatives (DCI), “Digital Cinema System Specification, Version 1.4” (see <https://www.dcinovies.com/specification/index.html>)