

AS/NZS IEC 62676.2.32:2020
IEC 62676-2-32:2019



Australian/New Zealand Standard™

Video surveillance systems for use in security applications

Part 2.32: Recording control and replay based on web services



AS/NZS IEC 62676.2.32:2020

This Joint Australian/New Zealand Standard™ was prepared by Joint Technical Committee EL-031, Intruder Alarm Equipment and Installations. It was approved on behalf of the Council of Standards Australia on 24 September 2020 and by the New Zealand Standards Approval Board on 7 October 2020.

This Standard was published on 23 October 2020.

The following are represented on Committee EL-031:

- Australian Digital and Telecommunications Industry Association
- Australian Industry Group
- Australian Security Industry Association
- Engineers Australia
- Fire Protection Association Australia
- Insurance Council of Australia
- NSW Police Force
- New Zealand Security Association
- Security Providers Association of Australia
- South Australia Police
- Victorian Security Institute

This Standard was issued in draft form for comment as DR AS/NZS IEC 62676.2.32:2020.

Keeping Standards up-to-date

Ensure you have the latest versions of our publications and keep up-to-date about Amendments, Rulings, Withdrawals, and new projects by visiting:

www.standards.org.au

www.standards.govt.nz

ISBN 978 1 76113 014 4

Australian/New Zealand Standard™

Video surveillance systems for use in security applications

Part 2.32: Recording control and replay based on web services

First published as AS IEC 62676.2.32:2020.



© IEC 2020 — All rights reserved

© Standards Australia Limited/the Crown in right of New Zealand, administered by the New Zealand Standards Executive 2020

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of either the IEC or the publisher, unless otherwise permitted under the Copyright Act 1968 (Cth) or the Copyright Act 1994 (New Zealand). If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please see the contact details on the back cover or the contact us page of the website for further information.

Preface

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee EL-031, Intruder Alarm Equipment and Installations.

The objective of this document is to specify the web service interface for the configuration of the recording of video, audio and metadata. Additionally, associated events are defined. Clause 4 provides a definition of the storage model this document is based on.

Web service usage is outside the scope of this document. Please refer to AS/NZS IEC 60839.11.31 for more information.

This document is identical with, and has been reproduced from, IEC 62676-2-32:2019, *Video surveillance systems for use in security applications — Part 2-32: Recording control and replay based on web services*.

As this document has been reproduced from an International Standard, the following applies:

- (a) In the source text “this part of IEC 62676” should read “this document”.
- (b) A full point substitutes for a comma when referring to a decimal marker.

Australian or Australian/New Zealand Standards that are identical adoptions of international normative references may be used interchangeably. Refer to the online catalogue for information on specific Standards.

The terms “normative” and “informative” are used in Standards to define the application of the appendices or annexes to which they apply. A “normative” appendix or annex is an integral part of a Standard, whereas an “informative” appendix or annex is only for information and guidance.

NOTES

CONTENTS

FOREWORD	6
INTRODUCTION.....	8
1 Scope	9
2 Normative references.....	9
3 Terms, definitions and abbreviated terms.....	10
3.1 Terms and definitions	10
3.2 Abbreviated terms	11
4 Overview.....	11
4.1 Interfaces.....	11
4.2 Storage model.....	12
4.3 Recording control	13
4.4 Search.....	14
4.5 Replay control.....	14
4.6 Export file format.....	14
4.6.1 Layout	14
4.6.2 Use case 1: Playback of chunked and oversize clips at remote site.....	15
4.6.3 Use case 2: Forensic analysis at court.....	16
4.6.4 Use case 3: Playback at players not equipped according to the present specification.....	16
4.7 Receiver	16
5 Recording control service.....	16
5.1 Overview	16
5.2 General requirements	18
5.3 Data structures.....	18
5.3.1 RecordingConfiguration	18
5.3.2 TrackConfiguration	18
5.3.3 RecordingJobConfiguration	18
5.4 CreateRecording	20
5.5 DeleteRecording	21
5.6 GetRecordings	21
5.7 SetRecordingConfiguration	22
5.8 GetRecordingConfiguration	22
5.9 CreateTrack	23
5.10 DeleteTrack	24
5.11 GetTrackConfiguration.....	24
5.12 SetTrackConfiguration	25
5.13 CreateRecordingJob	25
5.14 DeleteRecordingJob	26
5.15 GetRecordingJobs.....	27
5.16 SetRecordingJobConfiguration	27
5.17 GetRecordingJobConfiguration.....	28
5.18 SetRecordingJobMode.....	28
5.19 GetRecordingJobState.....	29
5.20 GetRecordingOptions	31
5.21 ExportRecordedData	31
5.22 StopExportRecordedData	32
5.23 GetExportRecordedDataState	33

5.24	GetServiceCapabilities	34
5.25	Events	35
5.25.1	General	35
5.25.2	Recording job state changes	35
5.25.3	Configuration changes	35
5.25.4	Data deletion	36
5.25.5	Recording and track creation and deletion	36
5.26	Examples	37
5.26.1	Example 1: setup recording of a single camera	37
5.26.2	Example 2: Record multiple streams from one camera to a single recording	38
6	Search service	38
6.1	General	38
6.2	Concepts	39
6.2.1	Search direction	39
6.2.2	Recording event	39
6.2.3	Search session	39
6.2.4	Search scope	40
6.2.5	Search filters	40
6.2.6	Time information	40
6.3	Data structures	40
6.3.1	RecordingInformation structure	40
6.3.2	RecordingSourceInformation structure	41
6.3.3	TrackInformation structure	41
6.3.4	SearchState Enumeration	42
6.3.5	MediaAttributes structure	42
6.3.6	FindEventResult structure	42
6.3.7	FindPTZPositionResult structure	42
6.3.8	PTZPositionFilter structure	42
6.3.9	MetadataFilter structure	43
6.3.10	FindMetadataResult structure	43
6.4	GetRecordingSummary	43
6.5	GetRecordingInformation	43
6.6	GetMediaAttributes	44
6.7	FindRecordings	45
6.8	GetRecordingSearchResults	45
6.9	FindEvents	46
6.10	GetEventSearchResults	47
6.11	FindPTZPosition	48
6.12	GetPTZPositionSearchResults	49
6.13	FindMetadata	50
6.14	GetMetadataSearchResults	51
6.15	EndSearch	52
6.16	GetServiceCapabilities	53
6.17	Recording event descriptions	53
6.18	XPath dialect	54
7	Replay control	55
7.1	Request replay URI	55
7.2	ReplayConfiguration	

7.3	SetReplayConfiguration	56
7.4	GetReplayConfiguration.....	56
7.5	GetServiceCapabilities	57
8	Playback	57
8.1	RTSP Usage	57
8.2	RTSP describe	58
8.3	RTP header extension	58
8.3.1	General	58
8.3.2	NTP timestamps.....	59
8.3.3	Compatibility with the JPEG header extension	59
8.4	RTSP feature tag.....	60
8.5	Initiating playback	60
8.5.1	General	60
8.5.2	Range header field	60
8.5.3	Rate-Control header field.....	61
8.5.4	Frames header field.....	61
8.5.5	Synchronization points.....	62
8.6	Reverse replay.....	62
8.6.1	Initiation.....	62
8.6.2	Packet transmission order.....	62
8.6.3	RTP sequence numbers.....	64
8.6.4	RTP timestamps.....	64
8.7	RTSP Keepalive	65
8.8	Currently recording footage.....	65
8.9	End of footage.....	65
8.10	Go To Time	65
8.11	Use of RTCP.....	65
9	Export file format	66
9.1	Required side information	66
9.2	Timing	68
9.3	Correction of start time	68
9.4	Signature	68
9.4.1	Preparing the signature input	68
9.4.2	Generating the signature	68
9.4.3	Include the generated signature in the file	69
9.5	Repeated signing	70
10	Receiver service	71
10.1	General	71
10.2	Synchronization points.....	72
10.3	Persistence	72
10.4	Receiver modes	72
10.5	Receiver commands	72
10.5.1	GetReceivers	72
10.5.2	GetReceiver	73
10.5.3	CreateReceiver	73
10.5.4	DeleteReceiver.....	73
10.5.5	ConfigureReceiver.....	74
10.5.6	SetReceiverMode	74
10.5.7	GetReceiverState	75

10.6	GetServiceCapabilitites	75
10.7	Events	76
10.7.1	General	76
10.7.2	ChangeState	76
10.7.3	Connection Failed	76
Annex A (informative)	Repeated signing	77
Annex B (normative)	Schema files	79
B.1	Recording control	79
B.2	Search	89
B.3	Replay control	96
B.4	Receiver	98
B.5	Common Schema	102
Bibliography		110
Figure 1	– Storage model with tracks	13
Figure 2	– Sealing and examination in a nutshell (Source: Wikipedia).....	15
Figure 3	– Example of recordings and tracks	17
Figure 4	– RecordingJobConfiguration structure	19
Figure 5	– RecordingJobStateInformation structure	30
Figure 6	– Recording state chart.....	41
Figure 7	– Packet transmission during forward playback	63
Figure 8	– Packet transmission during reverse playback	64
Figure A.1	– Single signature box arrangement	77
Figure A.2	– Repeated signature box arrangement	77
Table 1	– Referenced namespaces (with prefix)	12
Table 2	– Track configuration	21
Table 3	– RTP packet layout	58
Table 4	– RTP packet with JPEG header layout.....	59

INTERNATIONAL ELECTROTECHNICAL COMMISSION

VIDEO SURVEILLANCE SYSTEMS FOR USE IN SECURITY APPLICATIONS –

Part 2-32: Recording control and replay based on web services

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62676-2-32 has been prepared by IEC technical committee 79: Alarm and electronic security systems.

This first edition, together with IEC 60839-11-31 and IEC 62676-2-31, cancels and replaces IEC 62676-2-3:2013.

This edition includes the following significant technical changes with respect to IEC 62676-2-3:2013:

- a) an export file format has been added.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
79/621/FDIS	79/623/RVD

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62676 series, published under the general title *Video surveillance systems for use in security applications*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "<http://webstore.iec.ch>" in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

The goal of this document is to provide a fully interoperable network video recording and reply implementation comprised of products from different vendors. This document describes the network video recording model, interfaces, data types and data exchange patterns. The document reuses existing relevant standards where available, and introduces new specifications only where necessary to support the specific requirements for network video recording and reply.

VIDEO SURVEILLANCE SYSTEMS FOR USE IN SECURITY APPLICATIONS –

Part 2-32: Recording control and replay based on web services

1 Scope

This part of IEC 62676 specifies the web service interface for the configuration of the recording of video, audio and metadata. Additionally, associated events are defined.

Clause 4 provides a definition of the storage model this document is based on.

Web service usage is outside the scope of this document. Please refer to the IEC 60839-11-31 for more information

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60839-11-31:2016, *Alarm and electronic security systems – Part 11-31: Electronic access control systems – Core interoperability protocol based on Web Services*

IEC 62676-2-31:2019, *Video surveillance system for use in security applications – Part 2-31: Live streaming and control based on web services*

Internet Assigned Numbers Authority (IANA), Media Types, *Media Types* [online]. Edited N. Freed et al. [viewed 2019-02-28]. Available at <https://www.iana.org/assignments/media-types/media-types.xhtml>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 2326: *Real Time Streaming Protocol (RTSP)* [online]. Edited by H. Schulzrinne et al. April 1998 [viewed 2019-02-28]. Available at <http://www.ietf.org/rfc/rfc2326.txt>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 3280, *Internet X.509 Public Key Infrastructure – Certificate and Certificate Revocation List (CRL) Profile* [online]. Edited by Housley, et. al. April 2002 [Viewed 2019-02-28]. Available at <http://www.ietf.org/rfc/rfc3280.txt>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 3550, *RTP: A Transport Protocol for Real-Time* [online]. Edited by Schulzrinne, et al. Jul 2003 [viewed 2019-02-28]. Available at <https://www.ietf.org/rfc/rfc3550.txt>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 4055, *Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure – Certificate and Certificate Revocation List (CRL) Profile* [online]. Edited by Schaad, et al. June 2005 [viewed 2019-02-28]. Available at <https://www.ietf.org/rfc/rfc4055.txt>

The World Wide Web Consortium (W3C). *SOAP12-PART1, SOAP 1.2 – Part 1, Messaging Framework* [online]. Edited by M. Gudgin et al. Apr 2007 [Viewed 2019-02-28]. Available at <https://www.w3.org/TR/soap12-part1/>

The World Wide Web Consortium (W3C). XML-Schema 1, W3C XML Schema – Part 1: Structures Second Edition [online]. Edited by H. Thompson et al. Oct 2004 [viewed 2019-02-28]. Available at <https://www.w3.org/TR/xmlschema-1/>

The World Wide Web Consortium (W3C). XML-Schema 2, W3C XML Schema – Part 2: Datatypes Second Edition [online]. Edited by P. Biron et al. Oct 2004 [viewed 2019-02-28]. Available at <https://www.w3.org/TR/xmlschema-2/>

The World Wide Web Consortium (W3C). XPath 1.0, XML Path Language (XPath) Version 1.0 [online]. Edited by J. Clark et al. Nov 1999 [Viewed 2019-02-28]. Available at <https://www.w3.org/TR/1999/REC-xpath-19991116/>

Federal Information Processing Standard (FIPS), FIPS 180-4, *Secure Hash Standard (SHS)* [online]. [viewed 2019-02-28]
Available at <https://csrc.nist.gov/publications/detail/fips/180/4/final>

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1.1

access unit

one or more frames or samples of audio, video, or metadata, which are contained in a group of RTP packets having the same presentation time

3.1.2

certificate

data which binds a public key to a subject entity

Note 1 to entry: The certificate is digitally signed by the certificate issuer to allow for verifying its authenticity.

3.1.3

metadata

streaming data except video and audio, including video analytics results, PTZ position data and other metadata (such as textual data from POS applications)

3.1.4

recording

container for a set of audio, video and metadata tracks

Note 1 to entry: A recording can hold one or more tracks. A track is viewed as an infinite timeline that holds data at certain times.

3.1.5

recording event

event associated with a recording, represented by a notification message in the APIs

3.1.6**recording job**

job that performs the transfer of data from a data source to a particular recording using a particular configuration

3.1.7**signature**

digital signature

digital signature scheme

mathematical scheme for demonstrating the authenticity of a digital message or document

3.1.8**track**

individual data channel consisting of video, audio, or metadata

Note 1 to entry: This definition is consistent with the definition of "track" in IETF RFC 2326.

3.1.9**video analytics**

algorithms or programs used to analyse video data and to generate data describing object location and behaviour

3.2 Abbreviated terms

CCTV	closed-circuit television
JPEG	Joint Photographic Expert Group
PTZ	pan tilt zoom
RTCP	RTP control protocol
RTP	real-time transport protocol
RTSP	real time streaming protocol
SDP	session description protocol
SHA	secure hashing algorithm
TCP	transmission control protocol
UDP	user datagram protocol
UTC	coordinated universal time
UTF	Unicode Transformation Format
WSDL	web service description language

4 Overview**4.1 Interfaces**

This document provides a set of interfaces that enable the support of interoperable network storage devices, such as network video recorders, digital video recorders and cameras with embedded storage.

The following functions are supported:

- recording control;
- search;
- replay control.

These functions are provided by three interrelated services:

Recording control service enables a client to manage recordings, and to configure the transfer of data from data sources to recordings. Managing recordings includes creation and deletion of recordings and tracks. The WSDL for this service is specified in Clause B.1

Search service enables a client to find information about the recordings on the storage device, for example to construct a "timeline" view, and to find data of interest within a set of recordings. The latter is achieved by searching for events that are included in the metadata track recording. The WSDL for this service is specified in Clause B.2

Replay control service enables a client to play back recorded data, including video, audio and metadata. Functions are provided to start and stop playback and to change speed and direction of the replayed stream. It also enables a client to download data from the storage device so that export functionality can be provided. The WSDL for this service is specified in Clause B.3

This document also includes specification for:

- playback of recorded data;
- export file format;
- a receiver that acts as a RTSP client endpoint. The WSDL for this service is specified in Clause B.4.

Table 1 lists the prefix and namespaces used in this specification. For interfaces defined by this document, the respective annex is provided. Listed prefixes are not part of this document and an implementation can use any prefix.

Table 1 – Referenced namespaces (with prefix)

Prefix	Namespace URI	Reference
env	http://www.w3.org/2003/05/soap-envelope	W3C SOAP12-PART1
ter	http://www.onvif.org/ver10/error	IEC 60839-11-31
trc	http://www.onvif.org/ver10/recording/wsdl	Clause B.1
trp	http://www.onvif.org/ver10/replay/wsdl	Clause B.3
trv	http://www.onvif.org/ver10/receiver/wsdl	Clause B.4
tse	http://www.onvif.org/ver10/search/wsdl	Clause B.2
tt	http://www.onvif.org/ver10/schema	Clause B.5
xs	http://www.w3.org/2001/XMLSchema	W3C XML-Schema-1 W3C XML-Schema-2

4.2 Storage model

The storage interfaces in this document present a logical view of the data on the storage device. This view is completely independent of the way data might be physically stored on the disk.

The key concept in the storage model is that of a recording. The term recording is used in this specification to denote a container for a set of related audio, video and metadata tracks, typically from the same data source, for example a camera. A recording could hold any number of tracks. A track is viewed as an infinite timeline that holds data at certain times.

At a minimum, a recording is capable of holding three tracks, one for audio, one for video and one for metadata. Some implementations of the recording service can support multiple tracks of each type. For example, the same recording could hold two video tracks, one containing a low-resolution or low-frame-rate stream and one containing a high-resolution or high-frame-rate stream; see Figure 1.

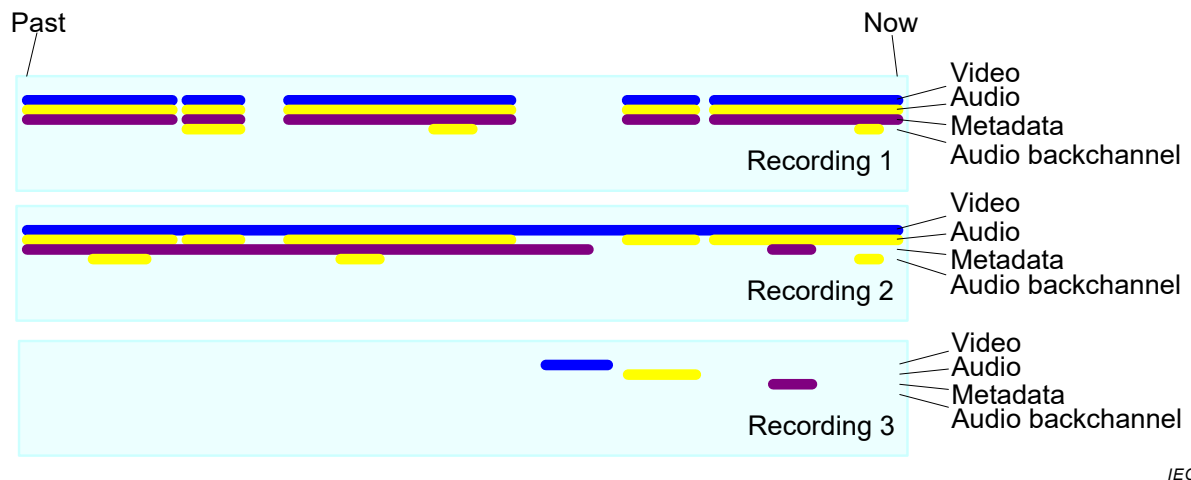


Figure 1 – Storage model with tracks

It is important to note that the storage interfaces do not expose the internal storage structures on the device. In particular, a recording is not intended to represent a single file on disk, although, in many storage device implementations, a recording is physically stored in a series of files. For instance, some camera implementations realise alarm recording by creating a distinct file for each alarm that occurs. Although each file could be represented as a different recording, the intent of the model in this document is that all these files are aggregated into a single recording.

Within a recording the regions where data is actually recorded are represented by pairs of events, where each pair comprises an event when recording started and an event when recording stopped. A client can construct the logical view of the recordings by using the FindRecordings and FindEvents methods of the search service.

If metadata is recorded, the metadata track can hold all the events generated by the data source, see Clause 10 of IEC 60839-11-31:2016 and 6.3.8 of IEC 62676-2-31:2019. In addition, a device also conceptually records historical events as defined by this document (see 6.17). This includes information such as the start and end of a recorded data range. A device can also conceptually record vendor specific historical events. Events generated by the device are not inserted in existing metadata tracks of recordings. The FindEvents method in the search service can find all the recorded events.

4.3 Recording control

The recording service enables a client to manage recordings, and to configure the transfer of data from data sources to recordings. Managing recordings includes the creation and deletion of recordings and tracks.

Recording jobs transfer data from a recording source to a recording. A recording source can be a receiver object created with the receiver service, or it can be a media profile that encodes data on a local device. The media profile could be used as a source on a camera with embedded storage.

To save data to a recording, a client first creates a recording and ensures that the recording has the necessary tracks. Then the client creates a recording job that pulls data from one or more sources and stores the data to the tracks in the recording.

Clients may set up multiple recording jobs that all record into the same recording. If multiple recording jobs are active, the device uses a priority scheme to select between the tracks defined in the recording jobs. Clients may change the mode of recording jobs at any time, thereby providing means to implement features like alarm recording or manual recording.

The recording job relies on the receiver service for receiving the data from other devices through receiver objects identified by ReceiverTokens.

4.4 Search

The search service enables a client to find information about the recordings on the storage device, for example to construct a "timeline" view, and to find data of interest within a set of recordings. The latter is achieved by searching for events and other information that is included in the metadata track recording.

The search service provides the following functionality:

- find recordings and information about each recording;
- find events in the metadata and among the historical events;
- find PTZ positions in the metadata;
- find other information in the metadata e.g. text from EPOS (electronic point-of-sale) systems.

The actual searching is done by coupled find and result operations and is asynchronous. Each find operation initiates a search session. The client can then acquire the results from the search session in increments, or all at once, depending on implementation and the scale of the search. There are four pairs of search operations for recordings, recording events, PTZ positions and metadata:

- FindRecordings and GetRecordingSearchResults;
- FindEvents and GetEventSearchResults;
- FindPTZPosition and GetPTZPositionSearchResults;
- FindMetadata and GetMetadataSearchResults.

4.5 Replay control

The replay control service provides a mechanism for the replay of stored video, audio and metadata. This mechanism can also be used to download data from the storage device so that export functionality can be provided.

The replay protocol is based on RTSP (IETF RFC 2326). However because RTSP does not directly support all of the requirements for replay, several extensions have been added to the protocol. In particular, an RTP header extension is defined to allow an absolute timestamp to be associated with each access unit (e.g. video frame), and to convey information about stream continuity.

The GetReplayUri command in the replay service returns the RTSP URL of a recording to allow it to be replayed using RTSP.

4.6 Export file format

4.6.1 Layout

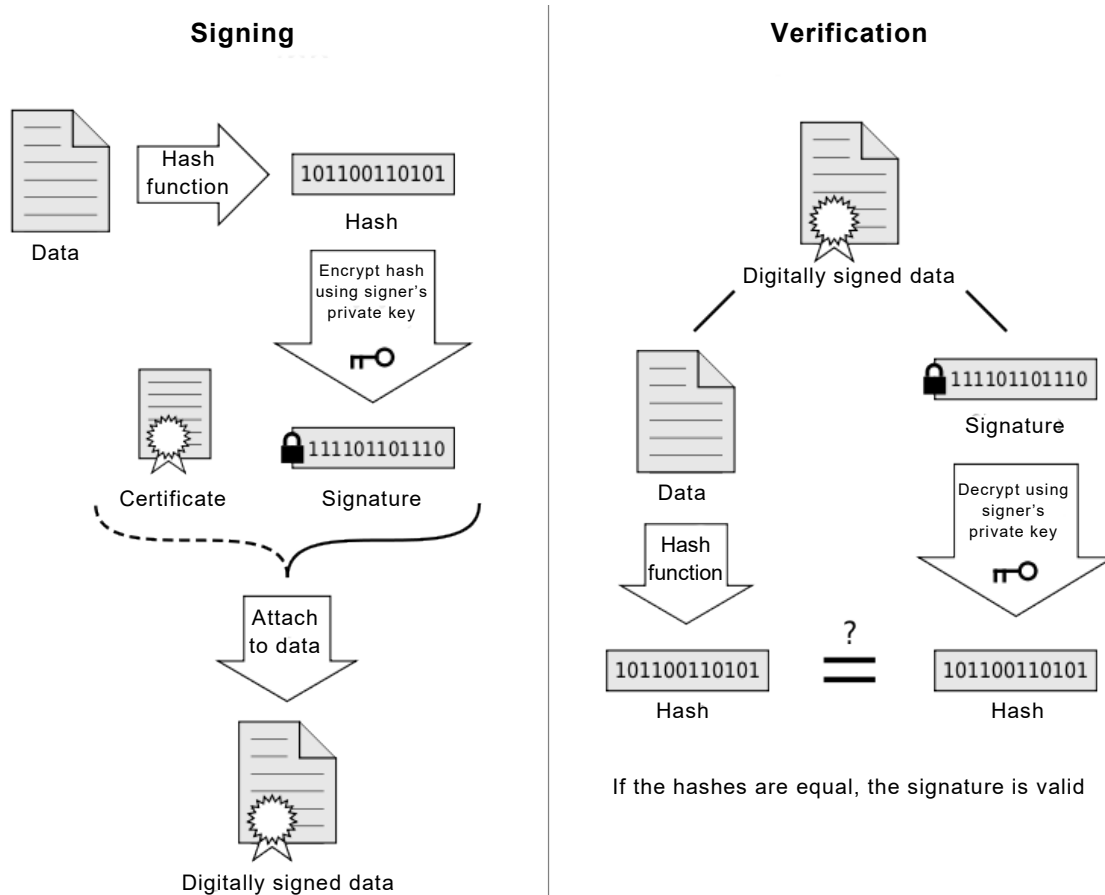
All data a user wishes to carry away separately are put into a metaphorical bag. The bag is then sealed to enable tamper detection. Anyone wanting to use the data from the bag first examines the seal. The data in the bag are identical with the original data as long as the seal is intact, see Figure 2.

Here, the metaphorical bag is represented by a file and the seal is represented by a signature over all data in the file.

The "bag of evidence" approach builds on procedures for media data and related metadata to be securely extracted from a trusted storage in a separate file. This approach defines which metadata have to be preserved in order to provide accurate replay. Data are provided "as is" without any further assertions, whatsoever, to perpetuate evidence.

Processing power reliefs include usage of hash functions before signature algorithms are applied. Multiple stages of signatures might be applied to collect additional information into a single sealed file.

International state of the art standards are applied for the file structure, hash and signature algorithms. The surveillance application format defined in ISO/IEC 23000-10 and the RSA2048 signature as well as the SHA-256 hash algorithm approved by NIST come into operation for the most widespread interoperability.



IEC

Figure 2 – Sealing and examination in a nutshell (Source: Wikipedia)

4.6.2 Use case 1: Playback of chunked and oversize clips at remote site

An operator exports a video clip with associated audio from a DVR of brand A onto two DVDs, because it didn't fit on one. The selected recording period contains gaps because the recorder only recorded when motion was detected. The DVD is then sent to a second site with software where the content of the DVDs is copied to the local hard disk. The user then plays it back in the video management system of brand B. The operator at the playback station wants to see the gaps in the recording, and seek any time where video has been exported. On playback, he expects the video to playback smoothly with lip sync audio.

4.6.3 Use case 2: Forensic analysis at court

A court receives video clips from a grocery store, a street surveillance system and a metro operator. All three videos are shown in the court's approved video player.

The judges want to see the suspect in all three video clips with exact time information. They also want to have information when the video clips have been exported and whether the video sequence is complete and authentic.

4.6.4 Use case 3: Playback at players not equipped according to the present specification

An authorized person receives video clips in the format defined in the present specification and wants to play back the media data on players conforming to the underlying standard definitions. Interpretation of the additional information added by the present specification is not required

4.7 Receiver

A receiver is an object that acts as an RTSP client endpoint. Receivers are used by other services that consume media streams, such as the recording and services. A receiver has a configuration that determines the RTSP endpoint to which it should connect and the connection parameters it should use.

A receiver can operate in three distinct modes:

- AlwaysConnect: the receiver attempts to maintain a persistent connection to the configured endpoint.
- NeverConnect: the receiver does not attempt to connect.
- AutoConnect: the receiver connects on demand, as required by consumers of the media streams.

A single receiver may be used by more than one consumer. For example, in order to record a stream and also perform analytics on it, both a recording job and an analytics engine could be attached to the same receiver. If the receiver uses the "Auto Connect" mode, it will connect whenever either the recording job or the analytics engine is active, and disconnect when neither of them are active.

Receivers may be created and deleted either manually, by calling the CreateReceiver and DeleteReceiver operations in the Receiver Service, or automatically by other services. For example, if a recording job is created with the "AutoCreateReceiver" option, it will automatically create and attach to a receiver. Deleting the recording job will also delete the receiver.

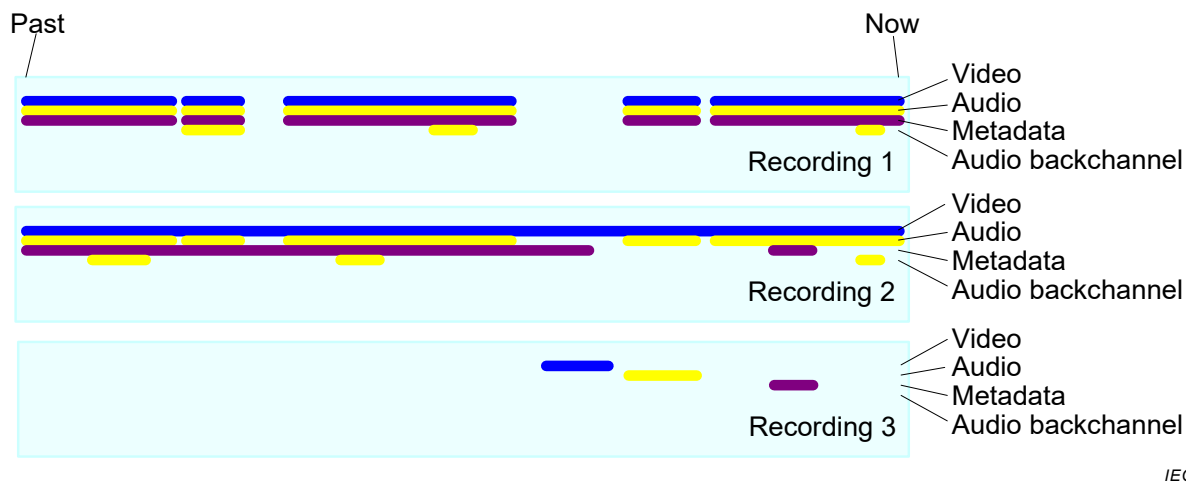
5 Recording control service

5.1 Overview

The recording service enables a client to manage recordings, and to configure the transfer of data from data sources to recordings. Managing recordings includes creation and deletion of recordings and tracks, as well as locking and unlocking ranges of recordings and deletion of recorded data.

Recording jobs transfer data from a recording source to a recording. A recording source can be a receiver object created with the receiver service, or it can be a media profile that encodes data on a local device. The media profile could be used as a source on a camera with embedded storage.

The term "recording" is used in this specification to denote a container for a set of audio, video and metadata tracks. A recording could hold any number of tracks. A track is viewed as an infinite timeline that holds data at certain times.



IEC

Figure 3 – Example of recordings and tracks

Figure 3 shows three recordings, each with a video, a metadata and two audio tracks. Here, a second audio track is used for storing the audio backchannel.

As a minimum, a recording shall be capable of holding three tracks: one for audio, one for video and one for metadata. Some implementations of the recording service can support multiple tracks of each type. All recorded data of a track shall have the same encoding.

To save data to a recording, a client first creates a recording and ensures that the recording has the necessary tracks. Then the client creates a recording job that pulls data from one or more sources and stores the data to the tracks in the recording.

Clients may set up multiple recording jobs that all record into the same recording. If multiple recording jobs are active, the device uses a priority scheme to select between the tracks defined in the recording jobs. Clients may change the mode of recording jobs at any time, thereby providing means to implement features such as alarm recording or manual recording.

The recording job relies on the receiver service for receiving the data from other devices through receiver objects identified by ReceiverTokens.

For the cases where a client uses a receiver object with a single recording job, the recording service can auto create and auto delete receiver objects. Autocreation is signalled with the AutoCreateReceiver flag in the recording job configuration structure. Receiver objects created this way shall be automatically deleted when no recording job uses them anymore. A receiver object that is automatically created shall have all its fields set to empty values. The client should configure the receiver object after it has created the recording job.

This document's view of recordings is a logical one, which is independent of the way recordings are physically stored on disk. For instance, some camera implementations realise alarm recording by creating a distinct file on a FAT file system for each alarm that occurs. Although each file could be represented as a different recording as defined by this document, the intent of the model in this document is that all these files are aggregated into a single recording. By searching for the "DataPresent" event with the FindEvents method of the search service, a client can locate the times at which video started to be recorded and where video stopped being recorded.

If metadata is recorded, the metadata track can hold all the events generated by the data source, see Clause 10 of IEC 60839-11-31:2016 and 6.3.8 of IEC 62676-2-31:2019. In addition, a device also conceptually record historical events as defined by this document, see 6.17, this includes information such as the start and end of a recorded data range. A device may also conceptually record vendor-specific historical events. Events generated by the device are not inserted in existing metadata tracks of recordings. The FindEvents method in the search service can find all the recorded events.

Many device implementations will automatically delete the oldest recorded data from storage in order to free up space for new recordings. Locks provide a mechanism to allow a user to select ranges of data. A range of data that is locked does not get deleted automatically. Support for locks is reserved for future versions of the specification.

5.2 General requirements

All the objects created within the recording service shall be persistent – i.e. they shall survive a power cycle. Likewise, all the configuration data in the objects shall be persistent.

5.3 Data structures

5.3.1 RecordingConfiguration

The RecordingConfiguration structure shall be used to configure recordings through CreateRecordings and Get/SetRecordingConfiguration.

MaximumRetentionTime specifies the maximum time that data in any track within the recording shall be stored. The device shall delete any data older than the maximum retention time. Such data shall not be accessible anymore. If the MaximumRetentionPeriod is set to 0, the device shall not limit the retention time of stored data, except by resource constraints. Whatever the value of MaximumRetentionTime, the device may automatically delete recordings to free up storage space for new recordings.

None of the other fields defined in this structure shall be used by the device. Instead, it simply stores this information, and it shall return it through the GetRecordingConfiguration and GetRecordingInformation methods.

A device may truncate any descriptive string without causing a fault if it exceeds the supported length. Descriptive strings are Location, Description and Content.

5.3.2 TrackConfiguration

The TrackConfiguration structure shall be used to configure tracks using CreateTrack and Get/SetTrackConfiguration.

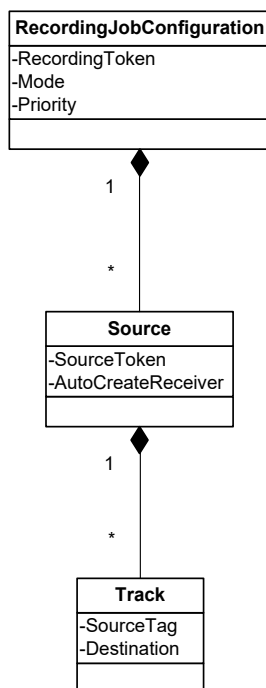
TrackConfiguration contains the following fields:

TrackType defines the data type of the track. It shall be equal to the strings "Video", "Audio" or "Metadata". The track shall only be able to hold data of that type.

None of the other fields defined in this structure shall be used by the device. Instead, it simply stores this information, and it shall return it through the GetTrackConfiguration and GetRecordingInformation methods.

5.3.3 RecordingJobConfiguration

The RecordingJobConfiguration structure shall hold the configuration for a recording job. As a UML diagram, the RecordingJobConfiguration can be viewed as shown in Figure 4



IEC

Figure 4 – RecordingJobConfiguration structure

The RecordingJobConfiguration holds the following fields:

RecordingToken: Identifies the recording to which this job shall store the received data.

Mode: If it is idle, nothing shall happen. If it is active and the recording job has the highest priority, the device shall try to obtain data from the receivers. A client shall use GetRecordingJobState to determine if data transfer is really taking place. The only valid values for Mode shall be "Idle" and "Active".

Priority: This shall be a non-negative number. If there are multiple recording jobs that store data to the same track, the device will only store the data for the recording job with the highest priority. The priority is specified per recording job, but the device shall determine the priority of each track individually. If there are two recording jobs with the same priority, the device shall record the data corresponding to the recording job that was activated last.

The value 0 indicates the lowest priority. Higher values shall indicate a higher priority.

SourceToken: This field shall be a reference to the source of the data. The type of the source is determined by the attribute Type in the SourceToken structure. The value is given in the form of an URI¹. This document defines the following two values for the attribute Type. If Type is <http://www.onvif.org/ver10/schema/Receiver>, the token is a ReceiverReference, see 10. In this case the device shall receive the data over the network. If Type is <http://www.onvif.org/ver10/schema/Profile>, the token identifies a media profile, see IEC 62676-11-31, instructing the device to obtain data from a profile that exists on the local device.

¹ The URI is only used to create a unique identifier and is not useful for retrieving any document.

A device that includes support for the media service as defined in IEC 62676-11-31 shall support a media profile token and a device that includes support for the receiver service as defined in this document shall support a Receiver token.

If the SourceToken is omitted, AutoCreateReceiver shall be true.

AutoCreateReceiver: If this field is TRUE, and if the SourceToken is omitted, the device shall create a receiver object (through the receiver service) and assign the ReceiverReference to the SourceToken field. When retrieving the RecordingJobConfiguration from the device, the AutoCreateReceiver field shall never be present.

SourceTag: If the received RTSP stream contains multiple tracks of the same type, the SourceTag differentiates between those Tracks.

Destination: The destination is the track token of the track to which the device shall store the received data. All tracks shall belong to the recording identified by the RecordingToken.

The TrackInformation field for a Track holds a single Source. In the case where multiple RecordingJobs with differing sources are recording to the same track, it is undefined which of them is reported in the corresponding TrackInformation of the RecordingSearch API.

5.4 CreateRecording

CreateRecording shall create a new recording. The new recording shall be created with a track for each supported TrackType (see 5.21).

This method is optional. It shall be available if the Recording/DynamicRecordings capability is TRUE.

REQUEST:

RecordingConfiguration [tt:RecordingConfiguration]

Contains the initial configuration for the recording.

RESPONSE:

RecordingToken [tt:RecordingReference]

The reference to the created recording.

FAULTS:

env:Receiver – ter:Action – ter:MaxRecordings

The device cannot create a new recording because it already has the maximum number of recordings that it supports.

env:Sender – ter:InvalidArgVal – ter:BadConfiguration

The RecordConfiguration is invalid.

env:Receiver – ter:ActionNotSupported – ter:NotImplemented

This optional method is not implemented.

ACCESS CLASS:

ACTUATE

When successfully completed, CreateRecording shall have created three tracks with the configurations shown in Table 2.

Table 2 – Track configuration

TrackToken	TrackType
VIDEO001	Video
AUDIO001	Audio
META001	Metadata

The RecordingConfiguration shall have the MaximumRetentionTime set to 0 (unlimited) and all TrackConfigurations shall have the Description set as an empty string.

5.5 DeleteRecording

DeleteRecording shall delete a recording object. Whenever a recording is deleted, the device shall delete all the tracks that are part of the recording, and it shall delete all the Recording Jobs that record into the recording. For each deleted recording job, the device shall also delete all the receiver objects associated with the recording job that are automatically created using the AutoCreateReceiver field of the recording job configuration structure and are not used in any other recording job.

This method is optional. It shall be available if the Recording/DynamicRecordings capability is TRUE.

REQUEST:

RecordingToken [tt:RecordingReference]

Identifies the recording that shall be deleted.

RESPONSE:

This is an empty message.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecording

RecordingToken does not reference an existing recording

env:Receiver – ter:ActionNotSupported – ter:NotImplemented

The device cannot delete recordings.

env:Receiver – ter:Action – ter:CannotDelete

This specific recording cannot be deleted.

ACCESS CLASS:

ACTUATE

5.6 GetRecordings

GetRecordings shall return a description of all the recordings in the device. This description shall include a list of all the tracks for each recording.

REQUEST:

This is an empty message.

RESPONSE:

RecordingItem – optional, unbounded [tt:GetRecordingsResponseItem]

Identifies a recording and its current configuration

FAULTS:

None

ACCESS CLASS:

READ_MEDIA

5.7 SetRecordingConfiguration

SetRecordingConfiguration shall change the configuration of a recording.

REQUEST:

RecordingToken [RecordingToken]

Identifies the recording that shall be changed.

RecordingConfiguration [RecordingConfiguration]

The new configuration for the recording.

RESPONSE:

This is an empty message.

FAULTS:

env:Sender – ter:InvalidArgVal – ter: BadConfiguration

The configuration is invalid.

env:Sender – ter:InvalidArgVal – ter:NoRecording

The RecordingToken does not reference an existing recording.

ACCESS CLASS:

ACTUATE

5.8 GetRecordingConfiguration

GetRecordingConfiguration shall retrieve the recording configuration for a recording.

REQUEST:

RecordingToken [tt:RecordingReference]

Identifies the recording for which the configuration shall be retrieved.

RESPONSE:

RecordingConfiguration [tt:RecordingConfiguration]

The current configuration for the requested recording.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecording

RecordingToken does not reference an existing recording.

ACCESS CLASS:

READ_MEDIA

5.9 CreateTrack

This method shall create a new track within a recording if the method GetRecordingOptions signals spare tracks for the recording. For a track to be created, the SpareXXX (where XXX is the track type) needs to be set.

This method is optional. It shall be available if the Recording/DynamicTracks capability is TRUE.

REQUEST:

RecordingToken [tt:RecordingReference]

Identifies the recording to which a track shall be added.

TrackConfiguration [tt:TrackConfiguration]

The configuration for the new track.

RESPONSE:

TrackToken [tt:TrackReference]

Identifies the newly created track. A device shall ensure that TrackToken is unique within the recording to which the new track belongs.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecording

RecordingToken does not reference an existing recording.

env:Receiver – ter:Action – ter:MaxTracks

The new track cannot be created because the maximum number of tracks that the device supports for this recording has been reached.

env:Sender – ter:InvalidArgVal – ter:BadConfiguration

TrackConfiguration is invalid.

env:Receiver – ter:ActionNotSupported – ter:NotImplemented

This optional method is not implemented.

ACCESS CLASS:

ACTUATE

A TrackToken in itself does not uniquely identify a specific track. Tracks within different recordings may have the same TrackToken.

5.10 DeleteTrack

DeleteTrack shall remove a track from a recording. All the data in the track shall be deleted.

This method is optional. It shall be available if the Recording/DynamicTracks capability is TRUE.

REQUEST:

RecordingToken [tt:RecordingReference]

Identifies the recording from which to delete the track.

TrackToken [tt:TrackReference]

Identifies the track to delete.

RESPONSE:

This is an empty message.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecording

RecordingToken does not reference an existing recording.

env:Sender – ter:InvalidArgVal – ter:NoTrack

TrackToken does not reference an existing track of the recording.

env:Receiver – ter:Action – ter:CannotDelete

This specific track cannot be deleted.

env:Receiver – ter:ActionNotSupported – ter:NotImplemented

This optional method is not implemented.

ACCESS CLASS:

ACTUATE

5.11 GetTrackConfiguration

GetTrackConfiguration shall retrieve the configuration for a specific track.

REQUEST:

RecordingToken [tt:RecordingReference]

Identifies the recording.

TrackToken [tt:TrackReference]

Identifies the track within the recording from which to get the track configuration

RESPONSE:

TrackConfiguration [tt:TrackConfiguration]

The current configuration for the track.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecording

RecordingToken does not reference an existing recording.

env:Sender – ter:InvalidArgVal – ter:NoTrack

TrackToken does not reference an existing track of the recording.

ACCESS CLASS:

READ_MEDIA**5.12 SetTrackConfiguration**

SetTrackConfiguration shall change the configuration of a track. TrackType shall be ignored by the device as it can't be changed. The TrackConfiguration is the new configuration for the track.

REQUEST:

RecordingToken [tt:RecordingReference]

Identifies the recording.

TrackToken [tt:TrackReference]

Identifies the recording within the recording from which to set the track configuration.

TrackConfiguration [tt:TrackConfiguration]

The new configuration for the track.

RESPONSE:

This is an empty message.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecording

RecordingToken does not reference an existing recording.

env:Sender – ter:InvalidArgVal – ter:NoTrack

TrackToken does not reference an existing track of the recording.

env:Sender – ter:InvalidArgVal – ter:BadConfiguration

The contents of the configuration object are invalid.

ACCESS CLASS:

ACTUATE**5.13 CreateRecordingJob**

CreateRecordingJob shall create a new recording job. A device shall support adding a RecordingJob to a recording for which it signals Spare jobs via GetRecordingOptions.

REQUEST:

JobConfiguration [tt:RecordingJobConfiguration]

The configuration of the new recording job.

RESPONSE:

JobToken [tt:RecordingJobReference]

Identifies the created recording job.

JobConfiguration [tt:RecordingJobConfiguration]

The configuration as it used by the device. This may be different from the JobConfiguration passed to CreateRecordingJob.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecording

The RecordingToken given in the JobConfiguration does not reference an existing recording.

env:Sender – ter:InvalidArgVal – ter:BadConfiguration

The contents of JobConfiguration are invalid.

env:Receiver – ter:Action – ter:MaxRecordingJobs

The maximum number of recording jobs that the device can handle has been reached.

env:Receiver – ter:Action – ter:MaxReceivers

If the AutoCreateReceivers flag is TRUE, this error can be returned if the receiver service cannot create a new receiver.

ACCESS CLASS:

ACTUATE

The JobConfiguration returned from CreateRecordingJob shall be identical to the JobConfiguration passed into CreateRecordingJob, except for the ReceiverToken and the AutoCreateReceiver. In the returned structure, the ReceiverToken shall be present and valid and the AutoCreateReceiver field shall be omitted.

5.14 DeleteRecordingJob

DeleteRecordingJob removes a recording job. It shall also implicitly delete all the receiver objects associated with the recording job that are automatically created using the AutoCreateReceiver field of the recording job configuration structure and are not used in any other recording job.

REQUEST:

JobToken [tt:RecordingJobReference]

Identifies the recording job that shall be deleted.

RESPONSE:

This is an empty message.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecordingJob
JobToken does not reference an existing job.

ACCESS CLASS:

ACTUATE**5.15 GetRecordingJobs**

GetRecordingJobs shall return a list of all the recording jobs in the device.

REQUEST:

This is an empty message.

RESPONSE:

JobItem– optional, unbounded [tt:GetRecordingJobsResponseItem]
Identifies a job in the device and holds its current configuration.

FAULTS:

None

ACCESS CLASS:

READ_MEDIA**5.16 SetRecordingJobConfiguration**

SetRecordingJobConfiguration shall change the configuration for a recording job. A device shall reject a request that tries to modify RecordingToken.

The JobConfiguration returned from SetRecordingJobConfiguration by a device shall be identical to the JobConfiguration passed into SetRecordingJobConfiguration, except for the ReceiverToken and the AutoCreateReceiver. In the returned structure, the ReceiverToken shall be present and valid and the AutoCreateReceiver field shall be omitted.

REQUEST:

JobToken [tt:RecordingJobReference]
Identifies the recording job to update.

JobConfiguration [tt:RecordingJobConfiguration]
The configuration to apply to the recording job.

RESPONSE:

JobConfiguration [tt:RecordingJobConfiguration]
The JobConfiguration structure shall be the configuration as it is used by the device. This may be different from the JobConfiguration passed to SetRecordingJobConfiguration.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecordingJob

JobToken does not reference an existing job.

env:Sender – ter:InvalidArgVal – ter:BadConfiguration

The contents of JobConfiguration are invalid.

env:Receiver – ter:Action – ter:MaxReceivers

If the AutoCreateReceivers flag is TRUE, this error can be returned if the receiver service cannot create a new receiver.

ACCESS CLASS:

ACTUATE

SetRecordingJobConfiguration shall implicitly delete any receiver objects that were created automatically if they are no longer used as a result of changing the recording job configuration.

5.17 GetRecordingJobConfiguration

GetRecordingJobConfiguration shall return the current configuration for a recording job.

REQUEST:

JobToken [tt:RecordingJobReference]

Identifies the recording job for which to retrieve the configuration.

RESPONSE:

JobConfiguration [tt:RecordingJobConfiguration]

The current configuration of the recording job.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecordingJob

JobToken does not reference an existing job.

ACCESS CLASS:

READ_MEDIA**5.18 SetRecordingJobMode**

SetRecordingJobMode shall change the mode of the recording job. Using this method shall be equivalent to retrieving the recording job configuration, and writing it back with a different mode.

Note that the state of a recording job will only become active if the recording job has the highest priority of all active jobs of a recording.

REQUEST:

JobToken [tt:RecordingJobReference]

Identifies the recording job for which to change the recording mode.

Mode [tt:RecordingJobMode]

The new mode for the recording job.

RESPONSE:

This is an empty message.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecordingJob

JobToken does not reference an existing job.

env:Sender – ter:InvalidArgVal – ter:BadMode

The Mode is invalid.

ACCESS CLASS:

ACTUATE**5.19 GetRecordingJobState**

GetRecordingJobState returns the state of a recording job. It includes an aggregated state, and state for each track of the recording job. The RecordingJobState may change due to

- calls that effect the RecordingJobMode, e.g. SetRecordingJobMode,
- internal recording engine state changes,
- changes in the recorded local media profile, or
- changes to the RTSP connection defined by the associated Receiver.

REQUEST:

JobToken [tt:RecordingJobReference]

Identifies the recording job for which to get the state.

RESPONSE:

State [tt:RecordingJobReference]

The state of the recording job.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecordingJob

JobToken does not reference an existing job.

ACCESS CLASS:

READ_MEDIA

The UML representation of the RecordingJobStateInformation structure is shown in Figure 5.

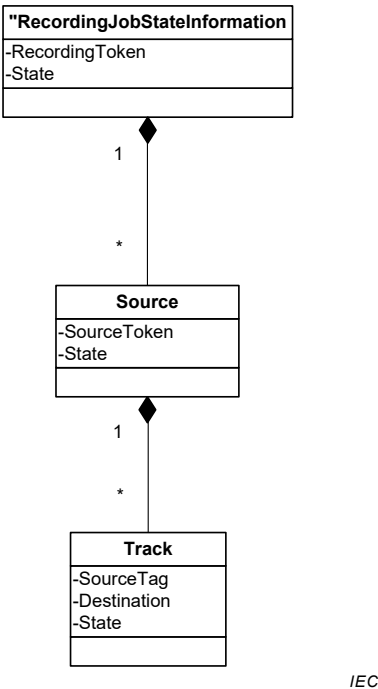


Figure 5 – RecordingJobStateInformation structure

RecordingToken shall be the identification of the recording that the recording job records to.

State (as part of RecordingJobStateInformation) shall hold the aggregated state over the whole RecordingJobInformation structure.

SourceToken shall identify the data source of the recording job.

State (as part of RecordingJobStateSource) shall hold the aggregated state over all substructures of RecordingJobStateSource.

SourceTag shall identify the track of the data source that provides the data.

Destination shall indicate the destination track

State (as part of RecordingJobTrackState) shall provide the job state of the track. The valid values of state shall be "Idle", "Active" and "Error". If state equals "Error", the Error field may be filled in with an implementation defined value.

Error, optional string describing the error state. The string should be in English. The following values are predefined:

"Incompatible Stream" – The stream cannot be recorded because the encoding does not match to previously recorded data.

A device shall apply the following rules to compute the aggregate state:

Idle	All state values in sub-nodes are "Idle"
PartiallyActive	The state of some sub-nodes are "Active" and some sub-nodes are "Idle"
Active	The state of all sub-nodes is "Active"
Error	At least one of the sub-nodes has the state "Error"

5.20 GetRecordingOptions

GetRecordingOptions returns information for a recording identified by the RecordingToken. The information includes the number of additional tracks as well as recording jobs that can be configured.

This method shall be supported if the Options support is signaled via the capabilities.

Note that this information is not static and is only guaranteed to be valid until the next modification of any recording jobs or tracks.

The track options shall be supported if the device signals support for dynamic tracks.

REQUEST:

RecordingToken [tt:RecordingReference]

Identifies the recording.

RESPONSE:

JobOptions [trc:JobOptions]

Contains two attributes:

Spare: Number of spare jobs that can be created for the recording. By setting none of the Spare attribute, the device signals that no job can be created.

CompatibleSources: A device that supports recording of a restricted set of media service profiles shall return the list of profiles that can be recorded on the given recording.

TrackOptions [trc:TrackOptions]

Contains four attributes:

SpareTotal: Total spare number of tracks that can be added to this recording.

SpareVideo: Number of spare video tracks for this recording

SpareAudio: Number of spare audio tracks for this recording

SpareMetadata: Number of spare metadata tracks for this recording

By setting none of the SpareXXX attributes, the device signals that no track can be added.

FAULTS:

env:Sender – ter:InvalidArgVal – ter:NoRecording

RecordingToken does not reference an existing recording.

ACCESS CLASS:

READ_MEDIA

5.21 ExportRecordedData

ExportRecordedData exports the selected recordings to the given storage target.

A device that indicates a capability of SupportedExportFileFormats shall support this command. For the parameter FileFormat, it shall accept any format that it advertises via the SupportedExportFileFormats capability.