ONVIF™
Export File Format Specification

Version 21.06

June, 2021

THIS DOCUMENT IS PROVIDED "AS IS," AND THE CORPORATION AND ITS MEMBERS AND THEIR AFFILIATES, MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

IN NO EVENT WILL THE CORPORATION OR ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT (1) THE CORPORATION, MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR (2) SUCH DAMAGES WERE REASONABLY FORESEEABLE, AND ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT. THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS AND THEIR RESPECTIVE AFFILIATES TO THE CORPORATION AND OTHER MEMBERS IN CERTAIN WRITTEN POLICIES OF THE CORPORATION.

CONTENTS

1 Scope

This document defines the ONVIF file format for exported media. The specification defines the mechanism necessary to support interoperable verification of the authenticity by the receiving party.

2 Normative references

ONVIF^{TM} Core Specification
<http://www.onvif.org/specs/core/ONVIF-Core-Specification.pdf>

ISO/IEC 14496-12 Information technology — Coding of audiovisual objects – Part 12: ISO base media file format
<https://www.iso.org/obp/ui/#iso:std:iso-iec:14496:-12:ed-5:v1:en>

ISO/IEC 23000-10 Information technology – Multimedia application format – Part 10: Surveillance application format
<https://www.iso.org/obp/ui/#iso:std:iso-iec:23000:-10:ed-2:v1:en>

ISO/IEC 23000-10/Cor 2:2014 Information technology – Multimedia application format – Part 10: Surveillance application format - TechnicalCorrigendum 2
<https://www.iso.org/obp/ui/#iso:std:iso-iec:23000:-10:ed-2:v1:cor:2:v1:en>

NIST FIPS 180-4 Secure Hash Standard
<https://csrc.nist.gov/publications/detail/fips/180/4/final>

ISO/IEC 14888-2 Information technology – Security techniques – Digital signatures with appendix – Part 2: Integer factorization based mechanisms
<https://www.iso.org/obp/ui/#iso:std:iso-iec:14888:-2:ed-2:v1:en>

PKCS#1, v2.1 RSA Cryptographic Standard

NIST FIPS 186 Digital Signature Standard (DSS)
<https://csrc.nist.gov/publications/detail/fips/186/4/final>

IETF RFC 3447 Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1
<https://tools.ietf.org/rfc/rfc3447.txt>

ITU-T Recommendation X.690 (2008) | ISO/IEC 8825-1:2008, Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER),Canonical Encoding Rules (CER)and Distinguished Encoding Rules (DER)
<https://www.itu.int/rec/T-REC-X.690-200811-S>

3 Terms and Definitions

3.1 Definitions

Certificate A certificate as used in this specification binds a public key to a subject entity. The certificate is digitally signed by the certificate issuer to allow for verifying its authenticity

Signature A digital signature or digital signature scheme is a mathematical scheme for demonstrating the authenticity of a digital message or document.

3.2 Abbreviations

SHA	Secure Hashing Algorithm

4 Overview

4.1 General

This specification extends the ISO/IEC 14496-12 Base File Format in order to serve Video Surveillance requirements.

4.2 Time Information

The ISO Base File Format has been mainly designed for storing movies and music clips. It defines a media timeline relative to the beginning which is defined as time zero. For Video Surveillance purposes it is important that the file preserves the absolute time of the captured frames. This specification refers to ISO 23000-10 for storing the absolute start time. All other times can be derived using the relative time data defined in ISO/IEC 14496-12. Additional to this time information time corrections can be stored during the sealing process.

In order to improve random access the ISOM baseline requires the movie fragment table at the end of the file. This redundant information does not require the protection seal.

4.3 Sealing

All data that 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. 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. It defines which metadata has to be preserved in order to provide for accurate replay. Data are provided “as is” without any further assertions, whatsoever, to perpetuate evidence.

Processing power usage can be reduced by performing 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 and the RSA2048 signature defined by ISO/IEC as well as the SHA-256 hash algorithm approved by NIST come into operation for most widespread interoperability.

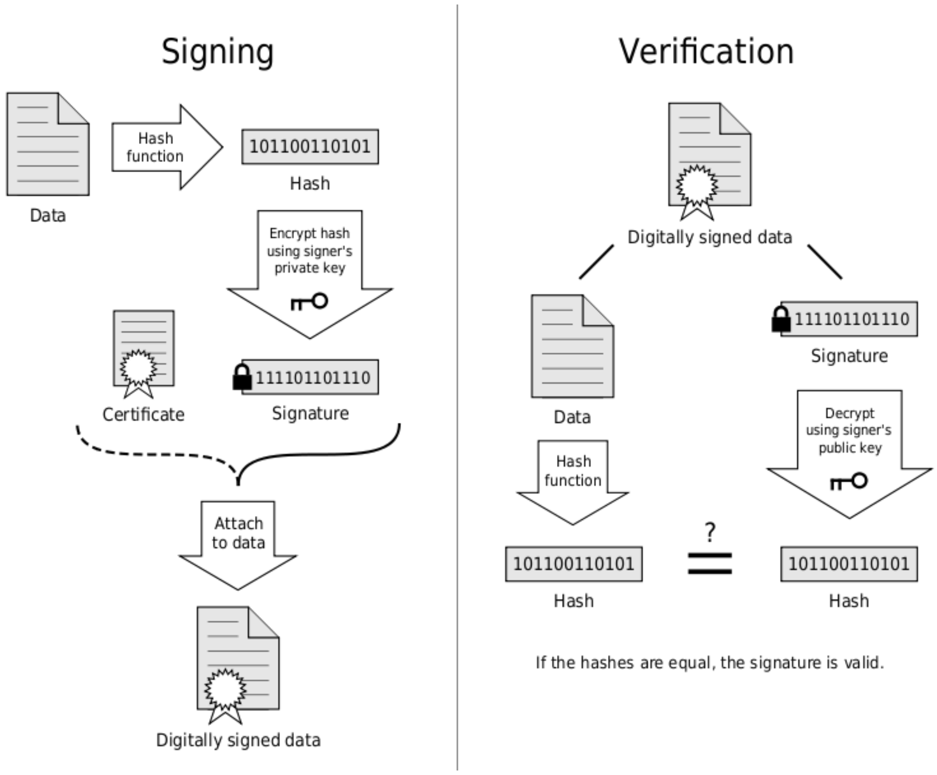

Figure 1: Sealing and examination process in a nutshell (Source: Wikipedia)

4.4 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 did only record when motion is 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 to seek to a time where Video has been exported. On playback he expects the Video to playback smoothly with lip sync Audio.

4.5 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 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 standards definitions. Interpretation of the additional information added by the present specification is not required.

5 Export Format

5.1 Required Side Information

The SurveillanceExportBox is required. It is recommended that the SurveillanceExportBox be placed as early as possible in files, for maximum utility.

In order to be able to associate the recording with a camera/microphone and the exporting system the following information shall be placed in the box:

- Source – Description of the video source
  - Name – Name of the camera
  - URL – Address under which the camera can be accessed
  - MAC – Unique physical address of the camera (examples: 08-00-27-00-0C-15, 08:00:27:00:0C:15, 080027000C15)
  - Line – Input line number token for multi channel devices
- Source – Description of the audio source
  - Name – Name of the microphone
  - URL – Address under which the microphone can be accessed
  - MAC – Unique physical address of the microphone
- Export – Unit executing the export
  - Name – Name of the exporting unit
  - URL – Address under which the exporting unit can be accessed
  - MAC – Unique physical address of the exporting unit

- ○ Time – Date and time information as to when the export was executed (start time)
- ○ Operator – Name or identification of the operator performing the export

SurveillanceExportBox

```
Box Type:  ‘suep’
Container: Meta Box (‘meta’), file level
Mandatory: Yes
Quantity:  Exactly one
```

This box shall contain information for all available tracks.

Syntax

```
class SurveillanceExportBox
  extends  FullBox(‘suep’, version = 1, 0){
  string   ExportUnitName;
  string   ExportUnitURL;
  string   ExportUnitMAC;
  UInt(64)   ExportUnitTime;
  string   ExportOperator;
  UInt(32) entry_count;
  int i;
  for (i=0; i < entry_count; i++) {
    UInt(16)   TrackID;
    string   SourceName;
    string   SourceURL;
    string   SourceMAC;
    string   SourceLine;
  }
}
```

Semantics

String items are null-terminated strings in UTF-8 characters. If not applicable, the string shall contain the null-termination only.

`ExportOperator` is a string that gives the name or identification of the operator performing the export. This string may be empty.

`ExportUnitTime` is an integer that provides date and time designation as defined in ISO/IEC 14496-12 of when the export operation has been started.

`entry_count` is an integer that provides the number of tracks.

5.2 Timing

The `startTime` element of `AFIdentificationBox`[1] shall contain the UTC based time of the first media sample in the fragment.

Each track fragment shall contain the Track Fragment Decode Time box ‘tfdt’ as defined in ISO/IEC 14496-12 to ease seeking during playback.

5.3 Correction of start time

CorrectStartTimeBox

```
Box Types: 'cstb'
```

[1] Box definitions can be found in ISO/IEC 23000-10 Information technology – Multimedia application format – Part 10: Surveillance application format.

```
Container: Protection Scheme Information Box (‘sinf’)
Mandatory: No
Quantity:  Zero or one per signing instance
```

Syntax

```
aligned(8)
class CorrectStartTimeBox extends Box (‘cstb’) {
  UInt(32) entry_count;
  for (i=0; i < entry_count; i++) {
    unsigned int(32) track_ID;
    unsigned int(64) startTime;
  }
)
```

Semantics

track_ID An integer that provides a reference to another track in the presentation. track_IDs are never re-used and cannot be equal to zero.

startTime The UTC based time represented by the number of 100-nanosecond intervals since January 1, 1601 of the first media sample in the first fragment.

5.4 Signature

5.4.1 Preparing the signature input

Inputs to the signature algorithm are all boxes of the file. These include boxes for signature creation, whose corresponding type is a string, set to a null value. The input contains signatures that are already present for repeated signing operations.

5.4.2 Generating the signature

Implementations of this specification shall support RSASSA-PSS signatures as specified in ISO/IEC 14888-2 and PKCS#1 v2.1 with:

- SHA-256 as specified in FIPS 180-4 as cryptographic hash function
- an RSA modulus length of at least 2048 bits
- MGF1 as specified in PKCS#1 v2.1 as mask generation algorithm with SHA-256 as cryptographic hash function
- Salt length 20
- Trailer field number as specified by the trailerFieldBC constant

Implementations may support other digital signature algorithms, if appropriate.

The generated signature string has to be included in the SignatureBox as defined in 5.4.3.

Generating and maintaining parameters of the signature algorithm, particularly signature and verification keys, is outside the scope of this document. Recommendations given, e.g., in FIPS 186 should be followed where appropriate.

5.4.3 Include the generated signature in the file

There are no changes to the file itself or the content after the signing operation has been performed. The sole exception is the input of the signature at the appropriate place.

The following box definitions provide for signature identification and inclusion. Encryption is not required; therefore an OriginalFormatBox is not necessary.

5.4.3.1 Item Protection Box

```
Box Type: 'ipro'[2]
Container: Meta box (‘meta’)
Mandatory: Yes
Quantity:  Exactly one
```

The `protection_count` shall be 1.

5.4.3.2 Protection Scheme Info Box

```
Box Type: 'sinf'[2]
Container: Item Protection Box (‘ipro’)
Mandatory: Yes
Quantity:  One per signing instance
```

Contains exactly one SchemeTypeBox and exactly one SchemeInformationBox.

5.4.3.3 Scheme Type Box

```
Box Type: 'schm'[2]
Container: Protection Scheme Information Box (‘sinf’)
Mandatory: Yes
Quantity:  One per signing instance
```

The `scheme_type` shall be 0x6F656666 („**O**nvif **E**xport **F**ile **F**ormat“).

The `scheme_version` shall be 0x00010000 (version 1).

5.4.3.4 Scheme Information Box

```
Box Type: 'schi'[2]
Container: Protection Scheme Information Box (‘sinf’)
Mandatory: Yes
Quantity:  One per signing instance
```

Contains exactly one SignatureBoxand exactly one CertificateBox. May also contain exactly one AdditionalUserInformationBox, exactly one SignatureConfigurationBox, and one CorrectStartTimeBox[3].

5.4.3.5 Signature Box

```
Box Type: 'sibo'
Container: Scheme Information Box (‘schi’)
Mandatory: Yes
Quantity:  One per signing instance
```

Syntax

```
aligned(8) class SignatureBox
extends Box(‘sibo’) {bit(8)  signature[];}
```

Semantics

`signature` binary byte array. Length depends on used RSA key length.

[2] Box definitions can be found in ISO/IEC 14496-12 Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format.

[3] CorrectStartTimeBox was added in version 1.1 of the ONVIF Export File Format

5.4.3.6 Certificate Box

```
Box Type: 'cert'
Container: Scheme Information Box (‘schi’)
Mandatory: Yes
Quantity:  One per signing instance
```

Syntax

```
aligned(8) class CertificateBox
extends Box(‘cert’) {
bit(8) data[];
}
```

Semantics

data is the DER encoded binary byte array representation of the certificate for the key that should be used to verify the signature in the SignatureBox

5.4.3.7 Signature Configuration Box

```
Box Type: 'sigC'
Container: Scheme Information Box (‘schi’)
Mandatory: No
Quantity:  Zero or one per signing instance
```

Syntax

```
aligned(8)
class SignatureConfigurationBox
  extends Box(‘sigC’) {
    bit(8)AlgorithmIdentifier[];
  }
```

Semantics

The ‘sigC’ box shall be present when the signature algorithm deviates from the default defined in 5.4.2. Its AlgorithmIdentifier is the signature algorithm identifier with optional parameters as defined by RFC 3280 and RFC 4055. It is encoded using the ASN.1 distinguished encoding rules (DER) and has the structure:

:AlgorithmIdentifier ::= SEQUENCE {

```
algorithm               OBJECT IDENTIFIER,
parameters              ANY DEFINED BY algorithm OPTIONAL
}
```

5.5 Repeated signing

5.5.1 Procedure

To add an item, for example, electronic receiving stamps, repeated signing of the file may be required.

Repeat steps defined in 5.4.1 and 5.4.2 and append another ProtectionSchemeInfoBox at the foot of the list of already existing boxes of that type as defined in 5.4.3 while not changing `protection_count` in the ItemProtectionBox. Parsers are required to check for the existence of multiple ProtectionSchemeInfoBox despite `protection_count` is fixed to 1, because any change of content which has already been signed would render the appropriate signature invalid. An optional AdditionalUserInformationBox might be used in order to add information.

In order to include optional user information, data related to an additional signature ‘auib’ box is provided.

5.5.2 Additional User Information Box

```
Box Type: 'auib'
Container: Scheme Information Box (‘schi’)
Mandatory: No
Quantity:  Zero or one per signing instance
```

Syntax

```
aligned(8)
class AdditionalUserInformationBox
  extends Box(‘auib’) {
    string  UserInformation;
  }
```

Semantics

`UserInformation` is a null terminated string in UTF-8 characters

Annex A. Repeated Signing (informative)

Figure A.1 characterizes the box arrangement defined in the present specification for data export.

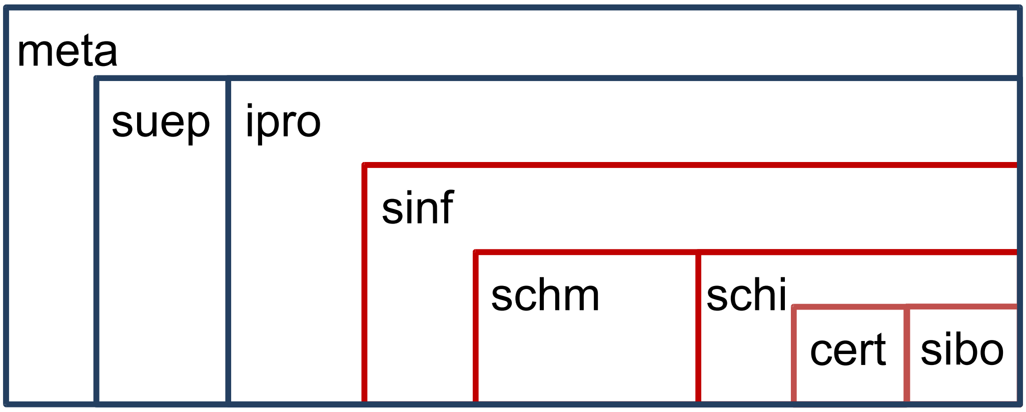

Figure A.1: Box structure with single signature

Figure A.2 characterizes the box arrangement after repeated signing.

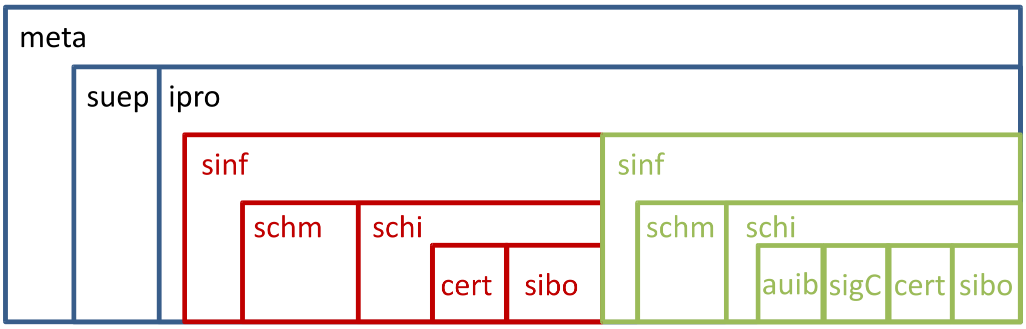

Figure A.2: Box structure with double signature

The red color represents the signature introduced at the export stage. The green color represents another signing operation happening after the export stage. The CertificateBox provides the public key for signature verification. Within the SignatureConfigurationBox information is contained describing a nondefault signature algorithm and its parameters. Additional information has been added in the AdditionalUserInformationBox.

In order to check validity of a signature the signature itself has to be taken from the SignatureBox and the bit values for the signature string be set to zero. The hashing operation is performed followed by the signature operation on the hash value. Now the two signatures can be compared.

Example: Steps to check validity of the first (red) signature from above

- Remove the (green) boxes created for the second signing
- Re-adjust box sizes of ‘ipro’ and ‘meta’ according to the size of removed ‘sinf’ box
- Read the public key from the red CertificateBox (do not change the box content)
- Take out the signature from the red SignatureBox
- Set the bit values of the red SignatureBox to zero

- Perform hash operation on the remaining file data
- Perform signature operation on the obtained hash value
- Compare the just generated signature with the signature taken out before

Annex B. Box Structure (informative)

The diagram below provides an overview on required boxes and their referencing for a Video stream with Audio. In order to simplify the example, it shows the content of a single fragment while a real file would contain numerous fragments.

```
ftyp[1]                    File brand ‘isom’.
moov[1]                    File wide definitions
    mvhd[1]                Movie header with creation time, timescale, duration and others
    trak[1]                First track is expected to contain Video
        tkhd[1]            Track header with creation time, timescale, duration, track ID
        mdia[1]
            mdhd[1]        Media header with creation time, timescale, duration and others
            hdlr[1]        Signals that this is a Video track (type is 'vide')
            minf[1]        Contains creation time, timescale, duration and others
                vmhd[1]    Video color information
                dinf[1]    Data location information.
                dref[1]
                    url[1] Data location flag in file must be set
            stbl[1]        Container with sample descriptions.
                stsd[1]    Codec information
                     avc1[1] H.264 codec information
                stts[1]    Sample index by time
                stsc[1]    Sample to chunk mapping
                stco[1]    List of Chunk offsets inside 'mdat' relative to file begin

    trak[1]                Second track with Audio
        tkhd[1]            Track header with creation time, timescale, duration, track ID
        mdia[1]
            mdhd[1]        Media header with creation time, timescale, duration and others
            hdlr[1]        Signals that this is a Audio track (type is 'soun')
            minf[1]        Contains creation time, timescale, duration and others
                mhd[1]     Audio stereo balance information
                dinf[1]    Data location information.
                dref[1]
                    url[1] Data location flag in file must be set
            stbl[1]        Container with sample descriptions.
                stsd[1]    Codec information
                    mp4a[1] Audio format information
                stts[1]    Sample index by time
                stsc[1]    Sample to chunk mapping
                stco[1]    List of Chunk offsets inside 'mdat' relative to file begin
mdat[1]                    Raw Video and Audio of first fragment (moov)
moof[1]                    Fragment
    mfhd[1]                Contains creation time, timescale, duration and others
    traf[1]                First track with Video
        tfhd[1]            Sample information
        tfdt[1]            Track fragment decode time
        trun[1]            Access to raw data in mdat box
    traf[1]                Second track with Audio
        tfhd[1]            Sample information
        tfdt[1]            Track fragment decode time
        trun[1]            Access to raw data in mdat box
mdat[1]                    Raw Video and Audio of this
meta[1]                    File level meta information
    hdlr[1]
    sumi[2]                File UUIDs as well as absolute start time and duration
    suep[3]                Export supplementary information
    ipro[1]                File protection
```

```
        sinf[1]             File protection  information
            schm[1]         Protection scheme OEFF defined by this specification
            schi[1]
                sibo[3]     Signature of the export
                cert[3]     Certificate of the exporter
mfra[1]                     Optional movie fragment random access (must be last in file)
    tfra[1]                 Track fragment random access
    mfro[1]                  Movie fragment random access offset
```

The superscripts denotes the specification that defines the box:

[1] ISO/IEC 14496-12

[2] ISO/IEC 23000-10

[3] This specification

Annex C. Revision History

Rev.	Date	Editor	Changes
1.0	March 2013	Gero Bäse	First release
1.0.1	May-2014	Michio Hirai	Change Request 1330
17.06	Jun-2017	Hans Busch, Hiroyuki Sano	Change Request 1843 Change Request 2065
18.06	Jun-2017	Stefan Andersson, Hans Busch	Add cstb box Add suep version 1 and Annex B
18.12	Dec-2018	Hiroyuki Sano	Change Request 2299, 2356, 2358, 2359, 2383, 2405
21.06	Jun-2021	Hans Busch	Move sigC definition to 5.4. Remove obsolete UUID notion.