

5

Digital Imaging and Communications in Medicine (DICOM)

Supplement ~~253~~: *Waveform Compression*

hat gelöscht: XXX

10

15

Prepared by: Working Group 32 Neurophysiology Waveforms

20 **DICOM Standards Committee, Working Group 6**

1812 N. Moore Street, Suite 2200

Arlington, VA 22209, USA

Status: Version 03, ~~Mar 18,~~ 2026 – ~~Before Public Comment~~

25 Developed pursuant to DICOM Work Item **2022-09-A**

hat gelöscht: 2

hat gelöscht: Jan

hat gelöscht: 4

hat gelöscht: First Read

Table of Contents

	Document History.....	4
	Open Issues.....	4
	Closed Issues.....	7
35	Scope and Field of Application.....	8
	Changes to NEMA Standards Publications PS3.2 Digital Imaging and Communications in Medicine (DICOM) Part 2: Conformance.....	9
	Changes to NEMA Standards Publications PS3.3 Digital Imaging and Communications in Medicine (DICOM) Part 3: Information Object Definitions.....	9
40	Changes to NEMA Standards Publications PS3.4 Digital Imaging and Communications in Medicine (DICOM) Part 4: Service Class Specifications.....	9
	C.6.1.1.5.1 Alternate Representation Sequence.....	9
	S.3.2.1.1.6 Allow Media Splitting.....	10
	Changes to NEMA Standards Publications PS3.5 Digital Imaging and Communications in Medicine (DICOM) Part 5: Data Structure and Semantics.....	10
45	3.10 DICOM Data Structures and Encoding Definitions.....	10
	8.2 Native or Encapsulated Format Pixel Data Encoding.....	11
	8.3 Waveform Data and related Data Elements.....	12
	8.x Native or Encapsulated Format Waveform Data Encoding.....	12
50	8.x.1 T.261 Waveform Compression.....	13
	A.4 Transfer Syntaxes for Encapsulation of Encoded Pixel Data.....	13
	A.X Transfer Syntaxes for Encapsulation of Encoded Waveform Data.....	13
	A.X.1 T.261 Lossless Transfer Syntax.....	14
	Changes to NEMA Standards Publications PS3.6 Digital Imaging and Communications in Medicine (DICOM) Part 6: Data Dictionary.....	14
55	Changes to NEMA Standards Publications PS3.10 Digital Imaging and Communications in Medicine (DICOM) Part 10: Media Storage and File Format for Media Interchange.....	15
	Changes to NEMA Standards Publications PS3.11 Digital Imaging and Communications in Medicine (DICOM) Part 11: Media Storage Application Profiles.....	15
60	Changes to NEMA Standards Publications PS3.12 Digital Imaging and Communications in Medicine (DICOM) Part 12: Media Formats and Physical Media for Media Interchange.....	15
	Changes to NEMA Standards Publications PS3.17 Digital Imaging and Communications in Medicine (DICOM) Part 17: Explanatory Information.....	15
	Changes to NEMA Standards Publications PS3.18 Digital Imaging and Communications in Medicine (DICOM) Part 18: Web Services.....	16
65	8.7.3.3.1 Uncompressed Bulk Data Media Types.....	18
	8.7.3.3.2 Compressed Bulk Data Media Types.....	19

Document History

2026/01/14	Version 1		Initial draft, First read
2026/03/14	Version 2		Updates during and after first part of First Read
2026/03/18	Version 3		WG-06-meeting
2026/03/19	Version 4		Updates during WG-32 meeting

hat gelöscht: r

70

Open Issues

1.	D.Clunie (WG-04, WG-06): Does it work for the use case (amount of compression achievable vs. other schemes (in the literature) in lossless and appropriate bit-rate lossy modes)?
3.	Q: Is the scheme patent and license fee free? A: The IP and patent issue was discussed with ITU/VCEG. ITU has similar rules as DICOM: if companies work in context of ITU they have to declare their patents, if there are any. The companies currently contributing to T.261 (Dolby, GE, and Philips) declared no patents right now. WG-06 will continue to check this with ITU at every milestone this supplement reaches in the DICOM standardization process.
5.	T.261 would allow to provide detailed timing information in the encoded bitstream. DICOM Waveforms do not support any timing information on samples (or parts of waveforms) besides AcquisitionDateTime (up to μ s precise) and the sampling frequency (Hz). There are attempts of some vendors to overcome physical hardware limitations resulting in subtle variations of the sampling frequency by providing detailed timing information on a per sample basis. Shall Transfer Syntaxes for Waveform compression make use of this feature even though this feature is not available in the uncompressed DICOM object at all? A: in principle yes, but more information about this feature is necessary → PS3.17 DICOM only supports synchronization mechanisms based on and related to the acquisition starting time. It should be considered whether and how this feature would affect synchronization with other objects or information (e.g. affect the precision of the location of annotations, although we do not expect the deviation being that large). 2026-01-15: Discussion in WG-32: * At least a timestamp per chunk in the offset table * Could this affect the synchronization of video? 2026-03-16: Discussion in WG-06: * If this is supported in the compressed object it will be lost in the decompressed. * Symmetry is preferred, so if this will be supported in the compressed object, it should also be provided in the uncompressed form. * How frequently is this used in existing implementations? * In MPEG, timing information is only kept in the video stream, and not exposed to the DICOM object.
6.	T.261 allows to store auxiliary metadata encoded in the bitstream, which are redundant to metadata provided in the DICOM Waveform: e.g. waveform type (EEG, EMG, ...). This

Kommentiert [SW1]: Check Frame Time / Frame Increment Pointer / Frame Time Vector

information does not affect the encoding or decoding process but are useful for non-DICOM applications.

A: in general: any metadata information should be separate from the compressed sample data – only clean sample data shall go to the encoded bitstream, nothing else.

Discussion of this issue:

2026-01-15: This requires further discussion with WG-04 / WG-06:

Does this mean “shall be stripped off”, “may be present but just not used”, ... ? incompatibilities between implementations of T.261 should be avoided.

2026-03-16 (WG-06):

* consider the transcoding situation: an application receives a T.261 encoded object containing these data and encapsulates this into a DICOM object – and the other way round.

* WG-06 prefers to have a clean codec not using any of these auxiliary metadata, even if these are separate parts of a codec: T.261 shall keep these data separable, one with metadata, one without – only the second shall be present in a DICOM object.

* if not stripping them off in transcoding, this might result in inconsistencies ...

* Hidden information inside of the encoded bitstream is a major issue in de-identification of DICOM objects.

* WG-32 2026-03-19: separation is no problem, it’s up to the implementation which packets to include in the encoded stream: DICOM has to state in its specification: **“certain packets shall not be present in the encoded stream”**

Kommentiert [SW2]: TODO: Check PS3.5 and how this is handled for jpeg/jpeg2000.

10. This supplement proposes a DICOM Transfer Syntax for lossless compression using T.261 codec using the default case “joint channels” – all channels of a Multiplex Group are considered as a single channel group.
Shall there also be a second DICOM Transfer Syntax for lossless compression allowing “individual channels” – each channel of a Multiplex Group goes into a separate channel group. This could result in better compression result, if single channels in a Multiplex Group are known to have little quality / much distortion.
A: WG-06 proposes to just have one lossless Transfer Syntax using the “joint channels”.
The “individual channels” mode is expected to result in little compression ratio.
For efficiency reasons this supplement should also address at least one lossy Transfer Syntax. The basic mechanism of encapsulation should work in the same way for both codecs. Considering both in this supplement ensures that encapsulation and compression/decompression works for both encodings.
- Discussion on this:
(Shouldn’t we also include the lossy TS – the details of the encoder will not go to the DICOM Standard: WG-06 assumes, that lossy compression / decompression might raise questions which would affect the encapsulation mechanism in general and which may not occur in the lossless case: e.g. effects of padding the “chunks”, creating discontinuities on decoding lossy encoded data, ...)
- 2026-01-15: timing is not affected at all by lossy / lossless, no effects on the encapsulation mechanism are to expect.
2026-02-20: the encoded “chunks” might be of odd length, padding might be an issue
No results on API discussion yet.
2026-03-16:
* WG-06 would prefer to work on the lossy and the lossless TS in one supplement.
* different TS means different mechanism, not only a different input parameter.

	<p><u>* if there is a broad variety of mechanism, the definition of the TS shall be rather narrow, in order to be clear for implementers.</u></p> <p><u>* think about default TS (explain the assoc. negot. Rules, accept header), if there is more than one (lossy) TS</u></p> <p>* WG-32 2016-03-19: only one profile is currently in T.261. No distinction of different lossy TSes is necessary. In principle the process is the same for lossless and for lossy, except the quality index.</p>
12.	<p>Q: According the current definition of Waveform IODs sample data are embedded in the metadata of the metadata (different to image instances, which have metadata in front of pixel data). It could be appropriate to shift the encoded bitstream to a new attribute located after the metadata <u>(and having sample data of different Multiplex Groups aligned, not separated by metadata)</u> (Re-writing the object is necessary in any case, e.g. to update the length information). <u>Would such an approach make it easier or faster to access the data?</u></p>
14.	<p>Q: Shall support of streaming be considered in context of this supplement?</p>
15.	<p>Q: Which encapsulation mechanism would be appropriate: having an Offset Table in the Sequence of encoded chunks, having an Offset Table as an Attribute in the metadata or have no offset table at all and leave accessing defined parts of the encoded bitstream to the decoder?</p> <p>Having the Offset Table as an Attribute in the metadata has turned out to be sometimes problematic: metadata may become quite large if there are many pointers in the Offset Table and some application store the whole set of metadata e.g. in a database field.</p> <p><u>Because the encapsulation mechanism for waveforms is new in the Standard,</u> the Offset Table as first item of the Sequence could use <u>64 bit pointers as well.</u></p> <p>Are there any arguments to have an OT at all?</p> <p><u>2026-03-16:</u></p> <p><u>* There are some arguments to have a separate OT in metadata (also e.g. with regard to the timing information, having separate OTs for different Multiplex Groups), but take care on the definition, so that these Attributes can be de-referenced and accessed as "bulk data" in DICOMweb (one of the problems with EOT in the past was the transformation of metadata to json in a DICOMweb application).</u></p> <p><u>see also #11</u> (→ Closed Issue) <u>see also #18</u></p>
16.	<p>Q: Shall support of encapsulated encoded waveform data be restricted to be only allowed for objects with a single Multiplex Group? What are the use cases (implementations) which make use of waveform objects with more than one Multiplex Group and are those such which would benefit from using the new Transfer Syntaxes?</p> <p><u>A: For the time being WG-32 is not aware of any use cases for EEG nor ECG which require more than one Multiplex Group.</u></p> <p><u>2026-03-26:</u></p> <p><u>* it would be preferable if the TS could be applied to ALL of the existing waveform SOP classes. Otherwise this would result in invalid TS/SOP class combination, which should be</u></p>

hat gelöscht: T

hat gelöscht: make

hat gelöscht: of the "large"

hat gelöscht: .

	<p><u>avoided.</u></p> <p><u>* Otherwise: remove the Multiplex Group Concept as a whole from the waveforms (which would be a major breaking change which requires investigations in the installed base (there are some vendors using DICOM ECG))</u></p>
18.	<p><u>Q: There might be different types of Offset Tables (32-bit pointers, 64-bit pointers, 64-bit pointers with chunk-timestamp), each as first item in the sequence of items holding the encoded bitstream. How should this be distinguished: a different Transfer Syntax, a different Data Element for the encapsulated data?</u></p> <p><u>2026-03-16:</u></p> <p><u>→ check also Frame Time Pointers in mpeg ...</u></p> <p><u>→ come back to this if we follow the concept of having the Offset Table as first item in the sequence.</u></p>

Closed Issues

2.	<p>Q: Is there open source reference software for encoding and decoding that can be tested and re-used in commercial implementations?</p> <p>A: Yes, https://vcgit.hhi.fraunhofer.de/vceg-sw/bwc</p>
4.	<p>Q: The parameters used in the encoding process are fully stored in the encoded bitstream. It is not necessary to apply them in the decoding process. Should these parameters nevertheless also be available in the DICOM metadata of the encoded waveform?</p> <p>A: No, not necessarily.</p> <p>Most important aspect is the consistency of metadata. Existing DICOM metadata affected by the encoding / decoding process shall be updated accordingly. Other metadata used internally by the codec shall not be duplicated to the DICOM metadata.</p>
7.	<p>Q: T.261 allows to store information about waveform features in the encoded bitstream, indicating that a portion of the waveform data has special characteristics or is of special interest. This information does not affect the encoding or decoding process but are useful for non-DICOM applications. Shall this feature be used in a DICOM context?</p> <p>A: No – see #6</p>
8.	<p>Q: T.261 allows to store annotation in the encoded bitstream. This information does not affect the encoding or decoding process but are useful for non-DICOM applications. Shall this feature of T.261 be used in a DICOM context?</p> <p>A: No – see #6. In DICOM objects annotations are stored using DICOM mechanisms (i.e. in the waveform object itself, in a Waveform Annotation SR or in a Waveform Presentation State).</p>
9.	<p>T.261 supports security features: providing user identification information, providing user authentication information, and providing authentication signature information: This information does not affect the encoding or decoding process but are required for non-DICOM applications. Shall this feature be used in a DICOM context?</p> <p>A: No – keep that separate. See #6.</p> <p>DICOM provides proper mechanisms for security and data integrity.</p>
11.	<p>There are two alternatives how the T.261 bitstream could be encapsulated in the DICOM object: one in analogy to compressed multiframe images with an offset table allowing to decompress individual “chunks”, the other in analogy to video Transfer Syntaxes without an offset table and just providing the whole bitstream. Does the encoded bitstream allow</p>

	decoding a specific portion of the waveform without storing the byte offsets in an offset table in the DICOM object? A: Yes.
17.	Q: Has the working group considered FLAC (RFC 9639 Free Lossless Audio Codec (FLAC) https://www.rfc-editor.org/rfc/rfc9639.html , https://en.wikipedia.org/wiki/FLAC) as one of the codec options? FLAC is an open specification with public tooling with a number of the relevant features for (lossless, good compression, block based, selectable block size, multi-channel, etc). A: FLAC was not evaluated because it has some significant limitations: it only supports up to 8 channels and it does lossless compression only. It is also not clear if there is sufficient maintenance for the codec by Xiph.org. But even though Xiph.org is not an international standards organization, FLAC could be considered in future if the encapsulation mechanism introduced in this supplement works in principle.

Scope and Field of Application

This supplement proposes to

- add an encapsulation mechanism for DICOM waveforms to encapsulate compressed data
- in analogy to the existing image compression and encapsulation of compressed pixel data.
- ~~A new Transfer Syntax for lossless compressed DICOM waveforms is added.~~
- ~~A new Transfer Syntax for lossy compressed DICOM waveforms is added.~~

This supplement adds Transfer Syntaxes for compression of DICOM Waveforms based on ITU/VCEG codec T.261, which was created especially for encoding and decoding of biomedical waveform data. This implies the introduction of an encapsulation mechanism for waveform sample data.

T.261 specifies an interoperable format for efficient compression, transmission, and storage of waveform signals, ~~including biomedical waveform signals such as electrocardiography (ECG), electroencephalography (EEG), electromyography (EMG), and photoplethysmogram (PPG) data, as well as other types of general waveform data.~~ It supports various bit depths, a broad range of sampling rates, and large numbers of channels, as required by a range of biomedical and general signal processing applications. T.261 provides for lossy, near-lossless, and lossless compression, and includes features such as optimized data blocking, indexing for rapid access ~~and~~ independent channel decoding, and metadata support.

~~A draft version of the specification of T.261 is available here:~~

~~https://www.itu.int/wftp3/av-arch/video-site/2601_Tel/VCEG-BZ02-H.BWC_Specification_Draft4-v1.docx~~

hat gelöscht: At least one

Kommentiert [SW3]: A single one? More than one?
→ check with ITU/WG-32

hat gelöscht: lossless

hat gelöscht: ,

hat gelöscht: including metadata enabling support of efficient artificial intelligence-based workflows...

hat formatiert: Schriftfarbe: Automatisch

105 **Changes to NEMA Standards Publications PS3.2**
Digital Imaging and Communications in Medicine (DICOM)
Part 2: Conformance

t.b.d.

110 **Changes to NEMA Standards Publications PS3.3**
Digital Imaging and Communications in Medicine (DICOM)
Part 3: Information Object Definitions

t.b.d. – any changes to IODs or modules (offset tables, lossy compression indication, ...)

115 **Changes to NEMA Standards Publications PS3.4**
Digital Imaging and Communications in Medicine (DICOM)
Part 4: Service Class Specifications

Modify C.6.1.1.5.1 as follows

[C.6.1.1 Patient Root Query Retrieve Information Model](#)

120 ...

C.6.1.1.5.1 Alternate Representation Sequence

125 The Alternate Representation Sequence (0008,3001) encodes a reference to an alternate encoding of the composite image **or waveform** identified in the query response item. This alternate encoding may utilize a different SOP Class or have different **image** quality characteristics, but it shall be the same image **or waveform**.

Note

The Alternate Representation Sequence (0008,3001) allows the query response about an original **image instance** to reference a lossy compressed version, and vice versa.

130 An **image instance** may be lossy compressed, e.g., for long-term archive purposes, and its SOP Instance UID changed. An application processing a SOP Instance that references the original **image instance** UID, e.g., a Structured Report, may query the C-FIND SCP for the **image instance**. The SCP returns a reference to an accessible version of the **image instance** even if the original SOP Instance is no longer available.

135 The Alternate Representation Sequence (0008,3001), if present in a query request Identifier, shall be zero-length, or shall contain a single zero-length Item. That is, only Universal Matching is defined for this Attribute.

The Alternate Representation Sequence (0008,3001), if present in the query response Identifier, may include zero or more Items.

Each Alternate Representation Sequence Item in the query response Identifier shall include

- 140 • the Series Instance UID (0020,000E) if the alternately encoded **image instance** is in a different Series.
- the SOP Class UID (0008,0016) and SOP Instance UID (0008,0018) of the alternately encoded **image instance**.
- 145 • the Purpose of Reference Code Sequence (0040,A170), which shall describe the nature of the alternate encoding of the **image instance**.
- The Purpose of Reference Code Sequence (0040,A170) shall include only one Item. The Baseline Context Group for this Code Sequence is CID 7205.

Modify S.3.2.1 Create a Media Creation Request as follows

150 **S.3.2.1 Create a Media Creation Request**

...

S.3.2.1.1.6 Allow Media Splitting

...

Note

- 155 1. If the file-set size exceeds the media storage capacity, and this flag has been set to NO, the SCP shall refuse to process the request.
- 2. If the requested Media Storage Application Profile allows for lossless compression, and images **or waveforms** are not already compressed, such compression may be applied by the SCP in order to fit all instances on a single piece of media. This also applies to lossy compression if it has not been allowed by the value of Allow Lossy Compression (2200,000F).
- 160

Modify S.3.2.1.1.9 Allow Lossy Compression as follows

...

165 The SCP shall not decompress and recompress already lossy compressed **images instances**, but may use **images instances** that have already been lossy compressed.

...

**Changes to NEMA Standards Publications PS3.5
Digital Imaging and Communications in Medicine (DICOM)
Part 5: Data Structure and Semantics**

170 *Update Section 2 Normative References as follows*

....

[ITU T.261] ITU. 2026 VCEG-BZ02-v1 H.BWC Draft 4 Specification Text

...

Update Section 3 Definitions as follows:

175 ...

3.10 DICOM Data Structures and Encoding Definitions

...

180 Encapsulated Format The Pixel Data Stream is encoded in a form wherein Fragments are contained within Item Tags within the Pixel Data (7FE0,0010) Data Element, usually used for compressed data.

185 Fragment One part of all of a Pixel Data Stream encoded within Encapsulated Pixel Data
Fragmentable Encapsulated Transfer Syntax A Transfer Syntax that allows the Encapsulated Format Pixel Data Stream to be split into one or more Fragments.

Native Format The Pixel Data Stream is encoded in an uncompressed form and occupies the entire Value of the Pixel Data (7FE0,0010) Data Element.

190 Non-Fragmentable Encapsulated Transfer Syntax A Transfer Syntax that requires the entire Encapsulated Format Pixel Data Stream be encoded in a single Fragment.

Pixel Data Stream The compressed or uncompressed octet-stream that constitutes the Pixel Data itself, without any DICOM-specific encapsulation structures.

195 Pixel Sample Value A value associated with an individual pixel. An individual pixel consists of one or more Pixel Sample Values (e.g., color images).

Waveform Data Stream The compressed or uncompressed octet-stream that constitutes the Waveform Data itself, without any DICOM-specific encapsulation structures.

200 **Waveform Chunk** One part of all of a Waveform Data Stream encoded with Encapsulated Waveform Data

Waveform Encapsulated Format The Waveform Data Stream is encoded in a form wherein Waveform Chunks are contained within Item Tags within the Waveform Data (5400,1010) Data Element, usually used for compressed data.

205 **Waveform Native Format** The Waveform Data Stream is encoded in an uncompressed form and occupies the entire Value of the Waveform Data (5400,1010) in a Multiplex Group.

210

Update Section 4 Symbols and Abbreviations as follows:

...

[ITU T.261] ITU. 2026 VCEG-BZ02-v1 H.BWC Draft 4 Specification Text

Kommentiert [SW5]: Discuss with David, if it would be okay to reuse Encapsulated Format and Native Format

215 *Update Section 8 Encoding of Pixel, Overlay and Waveform Data*

...
8.2 Native or Encapsulated Format Pixel Data Encoding

Modify Section 8.3 Waveform Data and Related Data Elements

220 **8.3 Waveform Data and related Data Elements**

The DICOM protocol provides for the exchange of encoded time-based signals, or waveforms, encoded in the Waveform Data (5400,1010).

Note:

- 225
1. Per Section 7.6, an IOD supporting multiple sets of Waveform Data will encapsulate Waveform Data (5400,1010) with a Sequence.
 2. **Multiple sets of Waveform Data originating from separate groups of channels are organized in Waveform Sequence (5400,0100), each representing a waveform multiplex group.**

Kommentiert [SW6]: Not clear what this refers to. Reference seems to be wrong, check if there is a section which mentions (5400,1010)

If Note 1. is to be removed, Note 2 shall become normal text.

230 Encoded Waveform Data of various bit depths is accommodated through the Waveform Bits Allocated (5400,1004) Data Element. This Data Element defines the size of each waveform data sample within the Waveform Data (5400,1010). Allowed Values are 8, 16, 32 and 64 bits.

The Value Representation of the Waveform Data (5400,1010) shall be OW; OB shall be used in cases where Waveform Bits Allocated has a Value of 8, but only with Transfer Syntaxes where the Value Representation is explicitly conveyed.

235 Note:

1. Under the Default Transfer Syntax, OB and OW VRs have the identical byte transfer order.
2. Conversion of a SOP Instance from the Default Transfer Syntax to an Explicit VR Transfer Syntax (uncompressed) requires the interpretation of the Waveform Bits Allocated (5400,1004) Data Element, to determine the proper VR of the Waveform Data.

240

The following Data Elements related to Waveform Data shall be encoded with the same VR as Waveform Data: Channel Minimum Value (5400,0110), Channel Maximum Value (5400,0112) and Waveform Padding Value (5400,100A).

245 *Add a new Chapter to Section 8 Encoding of Pixel, Overlay and Waveform Data*

8.x Native or Encapsulated Format Waveform Data Encoding

Waveform data conveyed in the Waveform Data Attribute (5400,1010) may be sent either in a Native (uncompressed) Format or in an Encapsulated Format (e.g., compressed).

< add restrictions to properties of Waveform Module for Native Waveform Data, if any >

250 If sent in an Encapsulated Format (i.e., other than the Native Format), the Value Representation OB is used. The Waveform Data Sample Values are encoded according to the encoding process defined by one of the negotiated Transfer Syntaxes (see Annex A) resulting in a set of byte streams.

Each Waveform Chunk conveys its own explicit even length (see Section A.X).

255 The Sequence of Waveform Chunks of the encapsulated stream is terminated by a Sequence Delimitation Item, thus allowing the support of encoding processes where the resulting length of the entire stream is not known until it is entirely encoded. Encapsulated Formats support Waveforms with a single Multiplex Group as well as Waveforms with multiple Multiplex Groups. At least one Multiplex Group shall be present and hence at least one Waveform Chunk will be present.

8.x.1 T.261 Waveform Compression

260 DICOM provides a mechanism for supporting the T.261 Waveform Compression through the Encapsulated Format. Annex A defines <> Transfer Syntaxes that reference the T.261 Standard and provide a lossless (bit preserving) and lossy compression schemes.

Note

265 The context where the usage of lossy compression of medical waveform data is clinically acceptable is beyond the scope of the DICOM Standard. The policies associated with the selection of appropriate compression parameters (e.g., compression ratio) for T.261 lossy compression is also beyond the scope of this Standard.

<add any restrictions, selection of default compression processes, if there are multiple TSes ...>

270 The use of the DICOM Encapsulated Format to support T.261 compressed Waveform Data requires that the Data Elements that are related to Waveform Data encoding (e.g.) shall contain values that are consistent with the characteristics of the compressed data stream.

<add required restrictions, limitations related to waveform properties ... >

275 10.1 DICOM Default Transfer Syntax

Modify Section A.4 Transfer Syntaxes for Encapsulation of Encoded Pixel Data

280

A.4 Transfer Syntaxes for Encapsulation of Encoded Pixel Data

285 These Transfer Syntaxes apply to the encoding of the entire DICOM Data Set, even though the image
Pixel Data (7FE0,0010) portion of the DICOM Data Set is the only portion that is encoded by an
encapsulated format. These Transfer Syntaxes shall only be used when Pixel Data (7FE0,0010) is
present in the top level Data Set, and hence shall not be used when Float Pixel Data (7FE0,0008) or
Double Float Pixel Data (7FE0,0009) are present. This implies that when a DICOM Message is being
290 encoded according to an encapsulation Transfer Syntax **for encoded Pixel Data** the following
requirements shall be met:

...

295

- **Waveform Data (5400,1010) has the Value Representation specified in its Explicit VR Field.
The component points shall be encoded in Little Endian.** (PS3.5 A.4 3. ...)

300

Add a new Section to Annex A Transfer Syntax Specifications (Normative)

A.X Transfer Syntaxes for Encapsulation of Encoded Waveform Data, These Transfer Syntaxes apply to
the encoding of the entire DICOM Data Set, even though the Waveform Data (5400,1010) portion of the
DICOM Data Set is the only portion that is encoded by an encapsulated format. These Transfer Syntaxes
shall only be used when Waveform Data (5400,0010) is present in the top level Data Set. This implies
305 that when a DICOM Message is being encoded according to an encapsulation Transfer Syntax for
encoded Waveform Data the following requirements shall be met:

1. The Data Elements contained in the Data Set structure shall be encoded with Explicit VR (with a
VR Field) as specified in Section 7.1.2.
2. The encoding of the overall Data Set structure (Data Element Tags, Value Length, etc.) shall be in
310 Little Endian as specified in Section 7.3.
3. The encoding of the Data Elements of the Data Set shall be as follows according to their Value
Representations:

....

- 315 • If Encapsulated Format, it has the Value Representation OB and is an octet-stream resulting from
one of the encoding processes. It contains the encoded Waveform Data Stream fragmented into
one or more Item(s). This Waveform Data Stream represents a single Multiplex Group.
- The Length of the Data Element (5400,0010) shall be set to the Value of Undefined Length
(FFFFFFFFH).
- 320 • Each Data Stream **Fragment** encoded according to the specific encoding process shall be
encapsulated as a DICOM Item with a specific Data Element Tag of Value (FFFE,E000). The
Item Tag is followed by a 4 byte Value (Item) Length Field encoding the explicit number of bytes
of the Item.

Kommentiert [SW7]: Can there be instances which contain pixel data AND Waveform Data?
I don't think so, proposal is to remove this statement.
Or express that this applies to Waveform Data contained in Image objects.

With sup98 the only object was retired (2004), which ever allowed having pixel data and waveform data in the same object.

hat gelöscht: ¶

- 325
- All items containing an encoded Waveform Chunk shall be made of an even number of bytes greater or equal to two and may be padded, if necessary, to meet the Sequence Item Format requirements of the DICOM Standard. ...
 - The first Item in the Sequence of Items before the encoded Waveform Data Stream shall be a Basic Offset Table item. The Basic Offset Table Item Value, however, is not required to be present:
- 330
- When the Item Value is not present, the Item Length shall be zero (0000000H).
 - When the Item Value is present, the Basic Offset Table Item Value shall contain concatenated 32-bit unsigned integer values that are byte offsets to the first byte of the Item Tag of each Chunk in the Sequence of Items. These offsets are measured from the first byte of the first Item Tag following the Basic Offset Table item.
- 335

Kommentiert [8]: See Open Issue #18

A.X.1 T.261 Lossless Transfer Syntax

< not defined yet >

340

**Changes to NEMA Standards Publications PS3.6
Digital Imaging and Communications in Medicine (DICOM)
Part 6: Data Dictionary**

Amend .Annex A / Table A-1. UID Values

345

Table A-1. UID Values

UID Value	UID Name	UID Keyword	UID Type	Part
...				
UID_TS1	T.261 Transfer Syntax for Lossless Waveform Compression	T.261Lossless	Transfer Syntax	PS3.5
...				

**Changes to NEMA Standards Publications PS3.10
Digital Imaging and Communications in Medicine (DICOM)
Part 10: Media Storage and File Format for Media Interchange**

350

t.b.d. ... e.g. DICOMDIR

**Changes to NEMA Standards Publications PS3.11
Digital Imaging and Communications in Medicine (DICOM)
Part 11: Media Storage Application Profiles**

355 *t.b.d. ... Profiles for specific data on media, e.g MPEG-4 on BD*

Kommentiert [SW9]: Question to WG-32:
Is there a need to specify a specific media support? Is it likely
that Waveform data are shared via media?
e.g. limits on filesizes on specific media (e.g. BD)

WG-06 on 18/03/2026: If a specific profile is needed, it shall
not be done in this supplement, but separately.

**Changes to NEMA Standards Publications PS3.12
Digital Imaging and Communications in Medicine (DICOM)
Part 12: Media Formats and Physical Media for Media Interchange**

360 *t.b.d. ... (e.g. contains the mime-type definition of application/dicom)*

**Changes to NEMA Standards Publications PS3.17
Digital Imaging and Communications in Medicine (DICOM)
Part 17: Explanatory Information**

365 *t.b.d.*

Length Limitations in DICOM

- 370
- The 32-bit Value Length field limits the maximum size of large data Value Fields such as Waveform Data or Pixel Data sent in Native Format (i.e. uncompressed)
 - compressed, the Pixel Data stream is fragmented into one or more Items in a Sequence of Items.
 - Each of these Items has a 4 byte (=32-bit Value Length field)
 - The first item in the Sequence is the Offset Table: contains 32-bit unsigned int values to the first byte of the first fragment for each frame.
- 375
- instead of the basic offset table an extended offset table might be used (Set of 64-bit pointer)
 - the Sequence itself (the encapsulation structure is of unlimited length, but limited by a Start Tag and an End Tag).

see also C.7.6.16.2.2.4 Concatenations and Stacks

380 Due to implementation specific reasons (such as maximum object size) ... e.g. for very large Multiframe Images multiple Instances are created, which together form a Concatenation ...

385 **Changes to NEMA Standards Publications PS3.18**
Digital Imaging and Communications in Medicine (DICOM)
Part 18: Web Services

390 *Modify 3.9 Web Service Definitions*

Bulk Data An object that contains an octet-stream containing one or more Value Fields (typically containing large data, such as Pixel Data) extracted from a DICOM Data Set. See Metadata.

Note:

1. The octet-stream does not include the Attribute Tag, Value Representation, or Attribute Length.
2. For the value of a Frame of a Pixel Data Attribute encoded in an Encapsulated Transfer Syntax, it does not include the Basic Offset Table and Pixel Data Stream Fragment Item Tags and lengths.
3. **For Waveform Data encoded in an Encapsulated Transfer Syntax ...**

...

400 Metadata A DICOM Data Set where zero or more elements (typically containing large data, such as Pixel Data **or Waveform Data**) have been replaced with Bulk Data URIs.

Note:

Metadata does not include the Group 0002 File Meta Information Data Elements, which describe but are not part of a Data Set, per Section 7.1 in PS3.10.

405 *Modify 8.7.1 DICOM Media Type Sets as follows*

...

410 If a Transfer Syntax parameter for a DICOM Media Type is not specified in a request or response, the Transfer Syntax in the response shall be the Transfer Syntax specified as the default for the Resource Category and Media Type combination in Table 8.7.3-2, Table 8.7.3-4 or Table 8.7.3-5, unless the origin server has only access to the pixel **or waveform** data in lossy compressed form or the pixel **or waveform** data in a lossless compressed or encapsulated uncompressed form **that is** of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.

...

415 Table 8.7.3-2, Table 8.7.3-3, Table 8.7.3-4, and Table 8.7.3-5 specify the Media Types used to encode different representations of DICOM Instances. These Media Types apply to all Resource Categories and have default encodings for images, **waveforms** and video data elements contained in the Instances.

Modify Table 8.7.2-1 Resource Categories as follows

Table 8.7.2-1. Resource Categories

Resource Category	Definition
-------------------	------------

Kommentiert [SW10]: Ask David ...
Consider rephrase this sentence
Put in front the main concept: unencodeable length ... and then describe the carious conditions

...	
Video	...
Waveform	This category includes all resources that are instances of a Waveform storage SOP Class.
Text	...
Other	This category includes all resources that are not included above, for example waveforms <t.b.d. something else> . (e.g. segmentation object)

420

Modify Table 8.7.3-2 Transfer Syntax UIDs for applicaion/dicom Media Types

Table 8.7.3-2. Transfer Syntax UIDs for application/dicom Media Types

Category	Transfer Syntax UID	Transfer Syntax Name	Optionality
...			
Video	...		
	1.2.840.10008.1.2.4.107	HEVC/H.265 Main Profile / Level 5.1	O
	1.2.840.10008.1.2.4.108	HEVC/H.265 Main 10 Profile / Level 5.1	O
Waveform	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
	UID TS1	T.261 Transfer Syntax for Lossless Waveform Compression	O
Text	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
Other	1.2.840.10008.1.2.1	Explicit VR Little Endian	D

425

Modify Section 8.7.3.3 Bulk Data Media Types as follows

...

430 The Selected Media Type will be the default Media Type for the Resource Category when the origin server supports none of the Acceptable Media Types, as described in Section 8.7.8, unless the origin server has only access to the pixel **or waveform** data in lossy compressed form or the pixel **or waveform** data in a lossless compressed or encapsulated uncompressed form **that**-is of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.

435 The Origin Server may support additional Transfer Syntaxes.

If no Media Type Transfer Syntax parameter is specified, then the Explicit VR Little Endian Transfer Syntax "1.2.840.10008.1.2.1" shall be used, unless the origin server has only access to the pixel **or waveform** data in lossy compressed form or the pixel **or waveform** data in a lossless compressed or encapsulated uncompressed form ~~that~~ is of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.

...

Depending on the Selected Media Type, the pixel data of a resource in the Video Category is encoded in:

- one Video compressed Bulk Data representation, or
- one Video uncompressed Bulk Data representation.

445

Depending on the Selected Media Type, the Waveform Data of a resource in the Waveform Resource Category is encoded in:

- **one Waveform compressed Bulk Data representation, or**
- **one Waveform uncompressed Bulk Data representation.**

450

8.7.3.3.1 Uncompressed Bulk Data Media Types

Table 8.7.3-4 specifies the default Media Type and Transfer Syntax UIDs, by Resource Category ...

Note

1.

455

...

Table 8.7.3-4. Transfer Syntax UIDs for Uncompressed Bulk Data

Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	RESTful
...				
Video	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
		1.2.840.10008.1.2.1.98	Encapsulated Uncompressed Explicit VR Little Endian	O
Waveform	<u>application/octet-stream</u>	<u>1.2.840.10008.1.2.1</u>	<u>Explicit VR Little Endian</u>	D
Text	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
Other	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D

...

460 **8.7.3.3.2 Compressed Bulk Data Media Types**

Compressed Bulk Data contains only the compressed octet stream without the fragment delimiters.

...

These Media Types can be used to retrieve Bulk Data, such as images, video, or waveform, encoded in a specific Transfer Syntax.

465 ...

Table 8.7.3-4. Media Types and Transfer Syntax UIDs for Compressed Bulk Data

Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	RESTful
...				
Video	video/mpeg	1.2.840.10008.1.2.100	MPEG2 Main Profile / Main Level	O
		1.2.840.10008.1.2.100.1	Fragmentable MPEG2 Main Profile / Main Level	O
		1.2.840.10008.1.2.101	MPEG2 Main Profile / High Level	D
		1.2.840.10008.1.2.101.1	Fragmentable MPEG2 Main Profile / High Level	O
	video/mp4	...		
	application/x-deflate	1.2.840.10008.1.2.8.1	Deflate Image Frame Compression	O
Waveform	t.b.d	UID TS1	T.261 Transfer Syntax for Lossless Waveform Compression	D
			Lossy ...	
Text		N/A (no defined compression transfer syntax for Text)		
Other		N/A (no defined compression transfer syntax for Other)		

Kommentiert [SW11]: Discuss this with WG-27. Ask Rob Horn and David via email ...