Digital Imaging and Communications in Medicine (DICOM)

Sup 244 - Frame Deflate Transfer Syntax

DICOM Standards Committee - Working Group 4 - Compression

1300 N. 17th Street Suite 1752
Rosslyn
VA
22209
USA

Status: Public Comment
Publication date 2024/03/19

This supplement is prepared pursuant to work item: 2023-10-A

Table of Contents

List of Tables

Document History

Document Version	Date	Content
01	2023/12/05	First draft for review by WG 4
02	2024/01/03	Draft for first read by WG 6
03	2024/01/09	After WG 6 review
04	2024/03/19	Public Comment Draft

To Do After Public Comment

1	Register new Content Type for Deflate if one is needed and IANA is willing.

Open Issues

1	There is no standard Content Type for deflated (RFC 1951) bit streams (probably because it is not usually used independently of gzip or zlib), so in the interim "application/vnd.dicom-deflate" is suggested. Alternatively, we could specify that gzip or zlib is used throughout, instead of deflate without a gzip or zlib wrapper; then the standard "application/gzip" or "application/zlib" could be used though these have some extra unnecessary stuff. See RFC 6713 and RFC 1952 and RFC 1950. This would be inconsistent with the existing (entire dataset) deflate transfer syntaxes, but that begs the question of why there is not an existing issue with this - the answer is that it is addressed in a note that says to use EVRLE with a Content-Encoding header of "deflate". WG 4 is seeking input from the community as to the best course of action.
2	Is the "deflate" value of Content-Encoding really the same as a deflated bit stream per RFC 1951 or is it actually using the zlib structure (defined in RFC 1950) with the deflate compression algorithm require to be used? If the latter then the existing note for the existing whole-dataset transfer syntax in PS3.18 is actually incorrect. Also, it is the entire multi-part stream to which the Content-Encoding applies, not the parts within it (???), though individual frames can still be retrieved as single part. There is also a problem with the Content-Encoding the entire PS3.10 file and not just the dataset after the PS3.10 meta information. WG 4 is again seeking input from the community as to the best course of action.
3	Should the frame deflate TS be added for Video to PS3.16 Table of Media Types and Transfer Syntax UIDs for Compressed Data in Bulkdata? The other image TS are not, so probably not, but input is sought on this question?

Closed Issues

1	The scope is limited to Pixel Data (7FE0,0010) since that is the primary use case (esp. for single bit Segmentation objects) and is satisfied with minimal changes. There is currently no Encapsulation mechanism defined for Float and Double Pixel Data, and extending support for that is beyond the scope of this Supplement. There is no provision proposed for selectively encoding other Data Elements than those related to Pixel Data that may be large and desirable to selectively retrieve in a compressed form, whether they be bulk data (like ICC Profiles, LUTs or contour data), or sequences (like Per-Frame Function Group Sequences) and extending that for that is also beyond the scope of this Supplement.

Scope and Field of Application

This Supplement adds a new Transfer Syntax primarily for single bit segmentation encoding, which is otherwise not well supported. There is a need to be able to store and transfer encoded single frames (such as for DICOMweb) rather than the entire dataset for those applications where only selected frames of a multi-frame object are required (such as for selected tiles at selected resolutions for whole slide images, or multi-organ segmentation of large volumetric CT or MR datasets).

PS3.2 DICOM PS3.2 - Conformance

Amend PS3.2 as follows (changes to existing text are bold and ***underlined*** *for additions and* ~~***struckthrough***~~ *for removals):*

Add new Transfer Syntax and Media Type to appropriate tables when finalized.

PS3.5 DICOM PS3.5 - Data Structures and Encoding

Amend PS3.5 as follows (changes to existing text are bold and ***underlined*** *for additions and* ***~~struckthrough~~*** *for removals):*

2 Normative References

[RFC1951] IETF. *DEFLATE Compressed Data Format Specification version 1.3*. http://tools.ietf.org/html/rfc1951 .

8 Encoding of Pixel, Overlay and Waveform Data

...

8.2 Native or Encapsulated Format Encoding

Pixel data conveyed in the Pixel Data (7FE0,0010) may be sent either in a Native (uncompressed) Format or in an Encapsulated Format (e.g., compressed).

...

Pixel Data conveyed in the Float Pixel Data (7FE0,0008) or Double Float Pixel Data (7FE0,0009) shall be in a Native (uncompressed) Format if encoded in a Standard Transfer Syntax.

Note

1. In future, if Standard Transfer Syntaxes are defined for compression of Float Pixel Data (7FE0,0008) or Double Float Pixel Data (7FE0,0009), this constraint may be relaxed and Encapsulated Format permitted.
2. This constraint does not apply to Private Transfer Syntaxes.

...

Float Pixel Data (7FE0,0008) is sent in Native Format; the Value Representation shall be OF, Bits Allocated (0028,0100) shall be 32, Bits Stored (0028,0101), High Bit (0028,0102) and Pixel Representation (0028,0103) shall not be present.

Double Float Pixel Data (7FE0,0009) is sent in Native Format; the Value Representation shall be OD, Bits Allocated (0028,0100) shall be 64, Bits Stored (0028,0101) and High Bit (0028,0102) and Pixel Representation (0028,0103) shall not be present.

...

If sent in an Encapsulated Format (i.e., other than the Native Format) the Value Representation OB is used. The Pixel Cells are encoded according to the encoding process defined by one of the negotiated Transfer Syntaxes (see Annex A).

...

8.2.n Deflated Image Frame Compression

DICOM provides a mechanism for supporting the use of Deflate compression of each individual image frame separately using the Encapsulated Format. Deflate is a byte oriented lossless compression scheme used with the Encapsulated Format.

Section A.4.n defines a Transfer Syntax that uses the Deflate algorithm defined in [RFC1951] to compress the pixel data of each image frame as a separate compressed bit stream.

Note

1. **Deflated Image Frame Compression is only applicable to pixel data conveyed in the Pixel Data (7FE0,0010) Data Element.**

2. **Though it uses the same compression scheme, the Deflated Image Frame Compression Transfer Syntax is distinct from the DICOM Deflated Little Endian Transfer Syntax (Explicit VR) Transfer Syntax defined in Section A.5, which compresses the entire Data Set as a single compressed bit stream.**

The use of the DICOM Encapsulated Format to support Deflated Image Frame Compression of Pixel Data requires that the Data Elements that are related to the Pixel Data encoding (e.g., Photometric Interpretation, Samples per Pixel, Planar Configuration, Bits Allocated, Bits Stored, High Bit, Pixel Representation, Rows, Columns, etc.) shall contain Values that are consistent with the compressed data.

Since the Deflate algorithm compresses byte streams regardless of their content and is not inherently specific to images, there are no restrictions on the values of Pixel Data Related Attributes.

Note

One application of Deflated Image Frame Compression is the lossless compression of bilevel (Bits Allocated (0028,0100) == 1) Segmentation images, but the mechanism is generic and not limited to any particular characteristics of the Pixel Data arrangement or values.

10 Transfer Syntax

...

10.1 DICOM Default Transfer Syntax

DICOM defines a Default Transfer Syntax, the DICOM Implicit VR Little Endian Transfer Syntax (identified by Transfer Syntax UID = "1.2.840.10008.1.2"), which shall be supported by every conformant DICOM Implementation. This implies that:

a. If an Application Entity issues an A-ASSOCIATE request, it shall offer the DICOM Implicit VR Little Endian Transfer Syntax in at least one of the Presentation Contexts associated with each offered Abstract Syntax.

Note

Offering Abstract Syntax (AS1) in two Presentation Contexts with Transfer Syntaxes (TS1) and (TS2) is not valid, but offering AS1-TS1, AS1-TS2 and AS1-TSD is valid because the DICOM Default Little Endian Transfer Syntax (TSD) is present in at least one of the Presentation Contexts that are based on Abstract Syntax (AS1).

b. If an Application Entity receives an A-ASSOCIATE indication corresponding to a request that follows the requirements specified in Section 10.1 (a), every Presentation Context related to a given Abstract Syntax cannot be rejected in an A-ASSOCIATE response for the reason that none of the Transfer Syntaxes are supported.

Note

When Abstract Syntax (AS1) is offered in three Presentation Contexts with Transfer Syntaxes (TS1), (TS2) and (TSD), the DICOM Default Little Endian Transfer Syntax (TSD) can be rejected if at least one of the other Presentation Contexts for Abstract Syntax (AS1) is accepted.

Both of these requirements, (a) and (b), are waived when the Application Entity sending the pixel data has only access to the pixel data in lossy compressed form or the pixel data in a lossless compressed or encapsulated uncompressed form that is of such length that it cannot be encoded in the Default Transfer Syntax, and a Transfer Syntax that uses a pixel data reference is not offered.

Requirement (b) to accept the Default Transfer Syntax is waived if a Transfer Syntax that uses a pixel data reference is offered.

Note

In other words, every sending AE is required to be able to convert any Data Set it is going to transmit into the Default Transfer Syntax, regardless of the form in which it originally received or stored the Data Set, except in the cases of when the decompressed or unencapsulated Pixel Data is too large to encode in the Default Transfer Syntax or is received in a lossy compressed form. In the case of lossy compressed Pixel Data, the sending AE is permitted to propose only the lossy compressed Transfer Syntax appropriate to the lossy form that was received. In the case of lossless compressed or encapsulated uncompressed Pixel Data that is too large to encode in the Default Transfer Syntax, the sending AE is permitted to propose any appropriate lossless compression Transfer Syntax or encapsulated uncompressed Transfer Syntax, not necessarily that in which the image was received, as an alternative to the Default Transfer Syntax.

This waiver does not apply to Data Sets received in a lossless compressed or encapsulated uncompressed form if the decompressed or unencapsulated Pixel Data is small enough to encode in the Default Transfer Syntax, which means that any AE receiving a Data Set in a lossless compressed Transfer Syntax or encapsulated uncompressed Transfer Syntax that needs to re-send the Data Set is required to be able to decompress or unencapsulate it in order to support (at least) the Default Transfer Syntax.

Similar concerns apply to the Web Services transactions and are addressed by specific requirements in PS3.18.

10.4 Transfer Syntax for DICOM RLE Image Compression

DICOM defines the RLE Image Compression (see ???). This implies that:

a. If an Application Entity issues an A-ASSOCIATE request where any offered Abstract Syntaxes is associated in one or more Presentation Contexts(s) with RLE compression Transfer Syntax, at least one of the Presentation Contexts that include this Abstract Syntax, shall include the DICOM Default Little Endian Transfer Syntax (uncompressed).

...

10.5 Transfer Syntax for a DICOM Default of Lossless and Lossy (Near-lossless) JPEG-LS Compression

One Transfer Syntax is specified for JPEG-LS Lossless Image Compression, and one Transfer Syntax is specified for JPEG-LS Lossy (Near-Lossless) Image Compression. The JPEG-LS Lossless Transfer Syntax shall be supported as a baseline if the JPEG-LS Lossy (Near-Lossless) Transfer Syntax is supported.

...

10.n Transfer Syntax for Deflated Image Frame Compression

One Transfer Syntax is specified for Deflated Image Frame Compression. There is no default or baseline specified (other than as described in Section 10.1).

a. **Since this is a lossless (reversible) Transfer Syntax, an Application Entity issues an A-ASSOCIATE request where any offered Abstract Syntaxes is associated in one or more Presentation Contexts(s) with Deflated Image Frame Compression Transfer Syntax, at least one of the Presentation Contexts that include this Abstract Syntax, shall include the DICOM Default Little Endian Transfer Syntax (uncompressed).**

A Transfer Syntax Specifications (Normative)

...

A.4 Transfer Syntaxes For Encapsulation of Encoded Pixel Data

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 encoded according to an encapsulation Transfer Syntax the following requirements shall be met:

1. ...
2. ...
3. The encoding of the Data Elements of the Data Set shall be as follows according to their Value Representations:
   - ...
   - For the Value Representations OB, OL, OV and OW, the encoding shall meet the following specification depending on the Data Element Tag:
     - Pixel Data (7FE0,0010) may be encapsulated or native.

It shall be encapsulated if present in the top-level Data Set (i.e., not nested within a Sequence Data Element).

> **Note**
>
> The distinction between defined Value Length (native) and undefined Value Length (encapsulated) is present so that the top level Data Set Pixel Data can be compressed (and hence encapsulated), but the Pixel Data within an Icon Image Sequence may or may not be compressed.

...

If encapsulated, it has the Value Representation OB and is an octet-stream resulting from one of the encoding processes. It contains the encoded pixel data stream fragmented into one or more Item(s). This Pixel Data Stream may represent a Single or Multi-frame Image. See ??? and ???.

- The Length of the Data Element (7FE0,0010) shall be set to the Value for Undefined Length (FFFFFFFFH).
- 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.

  > **Note**
  >
  > Whether more than one fragment per frame is permitted or not is defined per Transfer Syntax.

- All items containing an encoded fragment shall be made of an even number of bytes greater or equal to two. The last fragment of a frame may be padded, if necessary, to meet the sequence item format requirements of the DICOM Standard.

  > **Note**
  >
  > 1. Any necessary padding may be added in the JPEG or JPEG-LS compressed data stream as per ISO 10918-1 and ISO 14495-1 such that the End of Image (EOI) marker ends on an even byte boundary, or may be appended after the EOI marker, depending on the implementation.
  > 2. ISO 10918-1 and ISO 14495-1 define the ability to add any number of padding bytes FFH before any marker (all of which also begin with FFH). It is strongly recommended that FFH padding bytes not be added before the Start of Image (SOI) marker.
  > 3. **The end of a deflated bit stream will be indicated by the delimiter that will occur before any necessary padding with a single trailing NULL byte.**

- The first Item in the Sequence of Items before the encoded Pixel Data Stream shall be a Basic Offset Table item. The Basic Offset Table Item Value, however, is not required to be present:
  - When the Item Value is not present, the Item Length shall be zero (00000000H) (see ???).
  - 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 the first fragment for each frame in the Sequence of Items. These offsets are measured from the first byte of the first Item Tag following the Basic Offset Table item (see ???).

    > **Note**
    >
    > 1. For a Multi-Frame Image containing only one frame or a Single Frame Image, the Basic Offset Table Item Value may be present or not. If present it will contain a single 00000000H value.
    > 2. Decoders of encapsulated pixel data, whether Single Frame or Multi-Frame, need to accept both an empty Basic Offset Table (zero length) and a Basic Offset Table filled with 32 bit offset values.
    > 3. A Basic Offset Table Item Value is not permitted (i.e., the Item Length of the first Item will be zero) if Extended Offset Table (7FE0,0001) is present.

- This Sequence of Items is terminated by a Sequence Delimiter Item with the Tag (FFFE,E0DD) and an Value (Item) Length Field of Value (00000000H) (i.e., no Value Field shall be present).

- ...

...

A.4.2 RLE Image Compression

??? defines a RLE Image Compression Transfer Syntax. This Transfer Syntax is identified by the UID "1.2.840.10008.1.2.5". If the object allows multi-frame images in the pixel data field, then each frame shall be encoded separately. Each frame shall be encoded in one and only one Fragment (see Section 8.2).

...

A.4.11 Encapsulated Uncompressed Explicit VR Little Endian

The DICOM Transfer Syntax for Encapsulated Uncompressed Explicit VR Little Endian encodes a stream of one or more frames of uncompressed pixel data as Encapsulated fragments, and shall be identified by a UID of "1.2.840.10008.1.2.1.98".

If the object allows multi-frame images in the pixel data field, then each frame shall be encoded separately. Each frame shall be encoded in one and only one Fragment (see Section 8.2).

Within the Item Value of each Fragment (frame), the PixelData shall be encoded in the same manner as if it were encoded in Native format, including byte order, and padding to an even Item Length. OB VR shall be used, as required for all Encapsulated Format Transfer Syntaxes.

A.4.n Deflated Image Frame Compression

The DICOM Transfer Syntax for Deflated Image Frame Compression encodes a stream of one or more frames of compressed pixel data as Encapsulated Fragments. This Transfer Syntax is identified by a UID of "1.2.840.10008.1.2.1.uu".

If the object allows multi-frame images in the pixel data field, then each frame shall be encoded separately.

Each frame shall be encoded in one and only one Fragment (see Section 8.2).

The pixel data byte stream of each frame is separately compressed using the Deflate algorithm defined in [RFC1951].

If the Deflate algorithm produces an odd number of bytes then a single trailing NULL byte shall be added after the last byte of the deflated bit stream for each frame.

A.5 DICOM Deflated Little Endian Transfer Syntax (Explicit VR)

This Transfer Syntax applies to the encoding of the entire DICOM Data Set.

The entire Data Set is first encoded according to the rules specified in ???.

The entire byte stream is then compressed using the ~~"~~Deflate~~"~~ algorithm defined in ~~Internet RFC 1951~~**[RFC1951]**.

> **Note**
>
> **Though it uses the same compression scheme, the DICOM Deflated Little Endian Transfer Syntax (Explicit VR) Transfer Syntax is distinct from the Deflated Image Frame Compression Transfer Syntax defined in Section A.4.n, which compresses the pixel data of each image frame as a separate compressed bit stream and uses the Encapsulated Format Encoding.**

If the ~~d~~**D**eflate algorithm produces an odd number of bytes then a single trailing NULL byte shall be added after the last byte of the deflated bit stream.

> **Note**
>
> 1. The Pixel Data in Pixel Data (7FE0,0010), Float Pixel Data (7FE0,0008) or Double Float Pixel Data (7FE0,0009) is not handled in any special manner. The pixel data is first encoded as sequential uncompressed frames without encapsulation, and then is handled as part of the byte stream fed to the "deflate" compressor in the same manner as the Value Field of any other Data Element.

2. This Transfer Syntax is particularly useful for compression of objects without pixel data, such as structured reports. It is not particularly effective at image compression, since any benefit obtained from compressing the non-pixel data is offset by less effective compression of the much larger pixel data.

3. A freely available reference implementation of the "deflate" compressor may be found in the zlib package, which may be downloaded from http://www.zlib.net/.

4. Although the encoded stream may be padded by a trailing NULL byte, the end of the deflated bit stream will be indicated by the delimiter that will occur before the padding.

In order to facilitate interoperability of implementations conforming to the DICOM Standard that elect to use this Transfer Syntax, the following policy is specified:

- Any implementation that has elected to support the Deflated Explicit VR Little Endian Transfer Syntax for any Abstract Syntax, shall also support the Explicit VR Little Endian Transfer for that Abstract Syntax.

Note

1. This requirement to support the (uncompressed) Explicit VR Little Endian Transfer Syntax is in order to ensure full-fidelity exchange of VR information in the case that the Association Acceptor does not support the Deflated Explicit VR Little Endian Transfer Syntax. The requirement specified in Section 10.1 of this Part, that the Default Implicit VR Little Endian Transfer Syntax be supported by all implementations except those that only have access to lossy compressed pixel data, is not waived. In other words, an implementation must support all three Transfer Syntaxes.

2. There are no such "baseline" requirements on media, since such requirements are at the discretion of the Media Application Profile. Furthermore, sufficient object "management" information should be present in the DICOMDIR even if an individual application cannot decompress an instance encoded with the deflated Transfer Syntax.

This DICOM Deflated Explicit VR Little Endian Transfer Syntax shall be identified by a UID of Value "1.2.840.10008.1.2.1.99".

PS3.6 DICOM PS3.6 - Data Dictionary

Amend PS3.6 as follows (changes to existing text are bold and ***underlined*** *for additions and* ~~**struckthrough**~~ *for removals):*

A Registry of DICOM Unique Identifiers (UIDs) (Normative)

Table A-1. UID Values

UID Value	UID Name	UID Keyword	UID Type	Part
1.2.840.10008.1.2.1.99	Deflated Explicit VR Little Endian	DeflatedExplicitVRLittleEndian	Transfer Syntax	PS3.5
1.2.840.10008.1.2.1.uu	**Deflated Image Frame**	**DeflatedImageFrame**	**Transfer Syntax**	**PS3.5**

PS3.18 DICOM PS3.18 - Web Services

*Amend PS3.18 as follows (changes to existing text are bold and **underlined** for additions and **~~struckthrough~~** for removals):*

8.7.3.1 Instance Media Types

The application/dicom media type specifies a representation of Instances encoded in the DICOM File Format specified in Section 7 “DICOM File Format” in PS3.10.

Note

> The origin server may populate the PS3.10 File Meta Information with the identification of the Source, Sending and Receiving AE Titles and Presentation Addresses as described in Section 7.1 in PS3.10, or these Attributes may have been left unaltered from when the origin server received the objects. The user agent storing the objects received in the response may populate or coerce these Attributes based on its own knowledge of the endpoints involved in the transaction, so that they accurately identify the most recent storage transaction.

Table 8.7.3-2 specifies the default and optional Transfer Syntax UID combinations for each DICOM Resource Category (see ???). The default media type for the Resource Category shall be returned when the origin server supports none of the Acceptable Media Types, unless the origin server has only access to the pixel data in lossy compressed form or the pixel 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.

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

Category	Transfer Syntax UID	Transfer Syntax Name	Optionality
Single Frame Image	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
	1.2.840.10008.1.2.1.uu	**Deflated Image Frame**	**O**
	1.2.840.10008.1.2.4.70	JPEG Lossless, Non-Hierarchical, First-Order Prediction(Process 14 [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression	O
	1.2.840.10008.1.2.4.50	JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression	O
	1.2.840.10008.1.2.4.51	JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)	O
	1.2.840.10008.1.2.4.57	JPEG Lossless, Non-Hierarchical (Process 14)	O
	1.2.840.10008.1.2.5	RLE Lossless	O
	1.2.840.10008.1.2.4.80	JPEG-LS Lossless Image Compression	O
	1.2.840.10008.1.2.4.81	JPEG-LS Lossy (Near-Lossless) Image Compression	O
	1.2.840.10008.1.2.4.90	JPEG 2000 Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.91	JPEG 2000 Image Compression	O
	1.2.840.10008.1.2.4.92	JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.93	JPEG 2000 Part 2 Multi-component Image Compression	O
	1.2.840.10008.1.2.4.201	High-Throughput JPEG 2000 Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.202	High-Throughput JPEG 2000 with RPCL Options Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.203	High-Throughput JPEG 2000 Image Compression	O
Multi-frame Image	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
	1.2.840.10008.1.2.1.uu	**Deflated Image Frame**	**O**

Category	Transfer Syntax UID	Transfer Syntax Name	Optionality
	1.2.840.10008.1.2.4.90	JPEG 2000 Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.91	JPEG 2000 Image Compression	O
	1.2.840.10008.1.2.4.92	JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.93	JPEG 2000 Part 2 Multi-component Image Compression	O
	1.2.840.10008.1.2.4.201	High-Throughput JPEG 2000 Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.202	High-Throughput JPEG 2000 with RPCL Options Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.203	High-Throughput JPEG 2000 Image Compression	O
Video	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
	1.2.840.10008.1.2.1.uu	**Deflated Image Frame**	**O**
	1.2.840.10008.1.2.4.100	MPEG2 Main Profile @ Main Level	O
	1.2.840.10008.1.2.4.101	MPEG2 Main Profile @ High Level	O
	1.2.840.10008.1.2.4.102	MPEG-4 AVC/H.264 High Profile / Level 4.1	O
	1.2.840.10008.1.2.4.103	MPEG-4 AVC/H.264 BD-compatible High Profile / Level 4.1	O
	1.2.840.10008.1.2.4.104	MPEG-4 AVC/H.264 High Profile / Level 4.2 For 2D Video	O
	1.2.840.10008.1.2.4.105	MPEG-4 AVC/H.264 High Profile / Level 4.2 For 3D Video	O
	1.2.840.10008.1.2.4.106	MPEG-4 AVC/H.264 Stereo High Profile / Level 4.2	O
	1.2.840.10008.1.2.4.100.1	Fragmentable MPEG2 Main Profile @ Main Level	O
	1.2.840.10008.1.2.4.101.1	Fragmentable MPEG2 Main Profile @ High Level	O
	1.2.840.10008.1.2.4.102.1	Fragmentable MPEG-4 AVC/H.264 High Profile / Level 4.1	O
	1.2.840.10008.1.2.4.103.1	Fragmentable MPEG-4 AVC/H.264 BD-compatible High Profile / Level 4.1	O
	1.2.840.10008.1.2.4.104.1	Fragmentable MPEG-4 AVC/H.264 High Profile / Level 4.2 For 2D Video	O
	1.2.840.10008.1.2.4.105.1	Fragmentable MPEG-4 AVC/H.264 High Profile / Level 4.2 For 3D Video	O
	1.2.840.10008.1.2.4.106.1	Fragmentable MPEG-4 AVC/H.264 Stereo High Profile / Level 4.2	O
	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
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

Note

The Transfer Syntaxes used in a DICOM-RTV Metadata Flow are not included, since they are not used to produce a representation of an Instance encoded in the DICOM File Format.

8.7.3.3.2 Compressed Bulkdata Media Types

Compressed Bulkdata contains only the compressed octet stream without the fragment delimiters.

Table 8.7.3-5 specifies the default and optional media types and Transfer Syntax UID combinations for each Resource Category (see ???) of compressed Bulkdata for the RESTful services.

Note

1. Some of the Transfer Syntax Names include text about Default Transfer Syntax, however this applies to its role in DIMSE transactions, rather than the default for RESTful services (which is specified in the RESTful column of the table).
2. The Media Type column reflects the data encoding but does not include extended media type descriptors such as "multipart/related" that describe further packaging of the encoded data.

These media types can be used to retrieve Bulkdata, such as images or video, encoded in a specific Transfer Syntax.

For details on how Compressed Bulkdata is packaged into single part or multipart payloads, see ???.

Table 8.7.3-5. Media Types and Transfer Syntax UIDs for Compressed Data in Bulkdata

Resource Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	Optionality
Single Frame Image	image/jpeg	1.2.840.10008.1.2.4.70	JPEG Lossless, Non-Hierarchical, First-Order Prediction(Process 14 [Selection Value 1]) :Default Transfer Syntax for Lossless JPEG Image Compression	D
		1.2.840.10008.1.2.4.50	JPEG Baseline (Process 1) :Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression	O
		1.2.840.10008.1.2.4.51	JPEG Extended (Process 2 & 4) :Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)	O
		1.2.840.10008.1.2.4.57	JPEG Lossless, Non-Hierarchical (Process 14)	O
	application/vnd.dicom-deflate	**1.2.840.10008.1.2.1.uu**	**Deflated Image Frame**	**O**
	image/dicom-rle	1.2.840.10008.1.2.5	RLE Lossless	D
	image/jls	1.2.840.10008.1.2.4.80	JPEG-LS Lossless Image Compression	D
		1.2.840.10008.1.2.4.81	JPEG-LS Lossy (Near-Lossless) Image Compression	O
	image/jp2	1.2.840.10008.1.2.4.90	JPEG 2000 Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.91	JPEG 2000 Image Compression	O
	image/jpx	1.2.840.10008.1.2.4.92	JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.93	JPEG 2000 Part 2 Multi-component Image Compression	O
	image/jphc	1.2.840.10008.1.2.4.201	High-Throughput JPEG 2000 Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.202	High-Throughput JPEG 2000 with RPCL Options Image Compression (Lossless Only)	O
		1.2.840.10008.1.2.4.203	High-Throughput JPEG 2000 Image Compression	O
Multi-frame Image	image/jpeg	1.2.840.10008.1.2.4.70	JPEG Lossless, Non-Hierarchical, First-Order Prediction(Process 14 [Selection Value 1]) :Default Transfer Syntax for Lossless JPEG Image Compression	D
		1.2.840.10008.1.2.4.50	JPEG Baseline (Process 1) :Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression	O

Resource Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	Optionality
		1.2.840.10008.1.2.4.51	JPEG Extended (Process 2 & 4) :Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)	O
		1.2.840.10008.1.2.4.57	JPEG Lossless, Non-Hierarchical (Process 14)	O
	application/vnd.dicom-deflate	**1.2.840.10008.1.2.1.uu**	**Deflated Image Frame**	**O**
	image/dicom-rle	1.2.840.10008.1.2.5	RLE Lossless	D
	image/jls	1.2.840.10008.1.2.4.80	JPEG-LS Lossless Image Compression	D
		1.2.840.10008.1.2.4.81	JPEG-LS Lossy (Near-Lossless) Image Compression	O
	image/jp2	1.2.840.10008.1.2.4.90	JPEG 2000 Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.91	JPEG 2000 Image Compression	O
	image/jpx	1.2.840.10008.1.2.4.92	JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.93	JPEG 2000 Part 2 Multi-component Image Compression	O
	image/jphc	1.2.840.10008.1.2.4.201	High-Throughput JPEG 2000 Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.202	High-Throughput JPEG 2000 with RPCL Options Image Compression (Lossless Only)	O
		1.2.840.10008.1.2.4.203	High-Throughput JPEG 2000 Image Compression	O
Video	video/mpeg	1.2.840.10008.1.2.4.100	MPEG2 Main Profile @ Main Level	O
		1.2.840.10008.1.2.4.100.1	Fragmentable MPEG2 Main Profile @ Main Level	O
		1.2.840.10008.1.2.4.101	MPEG2 Main Profile @ High Level	D
		1.2.840.10008.1.2.4.101.1	Fragmentable MPEG2 Main Profile @ High Level	O
	video/mp4	1.2.840.10008.1.2.4.102	MPEG-4 AVC/H.264 High Profile / Level 4.1	D
		1.2.840.10008.1.2.4.102.1	Fragmentable MPEG-4 AVC/H.264 High Profile / Level 4.1	D
		1.2.840.10008.1.2.4.103	MPEG-4 AVC/H.264 BD-compatible High Profile / Level 4.1	O
		1.2.840.10008.1.2.4.103.1	Fragmentable MPEG-4 AVC/H.264 BD-compatible High Profile / Level 4.1	O
		1.2.840.10008.1.2.4.104	MPEG-4 AVC/H.264 High Profile / Level 4.2 For 2D Video	O
		1.2.840.10008.1.2.4.104.1	Fragmentable MPEG-4 AVC/H.264 High Profile / Level 4.2 For 2D Video	O
		1.2.840.10008.1.2.4.105	MPEG-4 AVC/H.264 High Profile / Level 4.2 For 3D Video	O
		1.2.840.10008.1.2.4.105.1	Fragmentable MPEG-4 AVC/H.264 High Profile / Level 4.2 For 3D Video	O
		1.2.840.10008.1.2.4.106	MPEG-4 AVC/H.264 Stereo High Profile / Level 4.2	O

Resource Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	Optionality
		1.2.840.10008.1.2.4.106.1	Fragmentable MPEG-4 AVC/H.264 Stereo High Profile / Level 4.2	O
	video/H265	1.2.840.10008.1.2.4.107	HEVC/H.265 Main Profile / Level 5.1	D
		1.2.840.10008.1.2.4.108	HEVC/H.265 Main 10 Profile / Level 5.1	O
Text		N/A (no defined compression transfer syntaxes for Text)		
Other		N/A (no defined compression transfer syntaxes for Other)		

The origin server may support additional Transfer Syntaxes.

For the media type image/jpeg Transfer Syntaxes, the image may or may not include the JFIF marker segment. The image may or may not include APP2 marker segments with an identifier of "ICC_PROFILE". There is no requirement for the origin server to add a JFIF marker segment nor to copy the value of the ICC Profile (0028,2000) Attribute, if present, into APP2 marker segments in the compressed data stream. See Section 8.2.1 “JPEG Image Compression” in PS3.5.

For the media type image/jp2 and image/jpx Transfer Syntaxes, the image does not include the jp2 marker segment. See Section 8.2.4 “JPEG 2000 Image Compression” in PS3.5 and Section A.4.4 “JPEG 2000 Image Compression” in PS3.5

Compressed multi-frame image pixel data is encoded as individual frames. E.g., each frame of a JPEG 2000 multi-frame image will be encoded separately as image/jp2 representations, rather than as a single video/mj2 (???) or application/octet-stream representation. See ??? for details on how multiple representations can be packaged into a multipart payload.

Video pixel data is encoded as a single video representation. E.g., all frames of an MPEG-4 video will be encoded as a single video/mp4 (???) representation.

Note

1. The resource on the origin server may have been encoded in the Deflated Explicit VR Little Endian (1.2.840.10008.1.2.1.99) Transfer Syntax. If so, the origin server may inflate it, and then convert it into an Acceptable Transfer Syntax. Alternatively, if the user agent allowed a Content-Encoding header field of 'deflate', then the deflated bytes may be transferred unaltered, but the Transfer Syntax parameter in the response should be the Explicit VR Little Endian Transfer Syntax.

   The resource on the origin server may have been encoded in the Deflated Image Frame (1.2.840.10008.1.2.1.uu) Transfer Syntax. If so, the origin server may return the compressed bit stream if it is an Acceptable Transfer Syntax, or the origin server may inflate it, and then convert it into an Acceptable Transfer Syntax. Alternatively, if the user agent allowed a Content-Encoding header field of 'deflate', then the deflated bytes may be transferred unaltered, but the Transfer Syntax parameter in the response should be the Explicit VR Little Endian Transfer Syntax.

2. Many of the media types used for compressed Pixel Data transferred as Bulkdata values are also used for consumer format media types. A web browser may not be able to display the encoded data directly, even though some of the same media types are also used for encoding rendered Pixel Data. See ???.

   For example, the media type for Bulkdata values of lossless 16-bit JPEG ??? encoded Pixel Data is "image/jpeg", the same media type as might be used for 8-bit JPEG ??? encoded Pixel Data, whether extracted as Bulkdata, or rendered. The Transfer Syntax parameter of the Content-Type header field is useful to signal the difference.

3. Previously, experimental Media Types "image/x-dicom-rle" and "image/x-jls" were defined, so origin servers and user agents may want to account for these when communicating with older implementations. These have been replaced with the standard Media Types "image/dicom-rle" and "image/jls", respectively.