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: Draft for Letter Ballot
Publication date 2024/08/20

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
05	2024/07/06	After Public Comment
06	2024/07/30	Draft for review by WG 04 before WG 06
07	2024/08/19	Draft for review by WG 06 before Letter Ballot
08	2024/08/20	Letter Ballot

To Do

1	Register new Content Type for application/deflate - submitted as IANA #1367781 - have provisional registration in http://www.iana.org/assignments/provisional-standard-media-types - awaiting final response from reviewer after 2nd round of answers to their questions.
2	Rationalize use of "bit stream"vs. "byte stream".

Open Issues

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.
2	The "deflate" value of Content-Encoding is not really the same as a deflated bit stream per RFC 1951, since officially it actually requires using the zlib structure (defined in RFC 1950) with the deflate compression algorithm require to be used. Per the current RFC 7230 HTTP/1.1 Message Syntax and Encoding Section 4.2.2 Deflate Coding '*The "deflate" coding is a "zlib" data format [RFC1950] containing a "deflate" compressed data stream [RFC1951] that uses a combination of the Lempel-Ziv (LZ77) compression algorithm and Huffman coding. Note: Some non-conformant implementations send the "deflate" compressed data without the zlib wrapper.*'. The clarifying note has apparently been added to the earlier description in RFC 2616 HTTP/1.1 Section 3.5 Content Encodings. The existing PS3.18 note for the existing whole-dataset transfer syntax is therefore incorrect in this respect. Though it could be amended to match the RFC requirement for the zlib container, as is done for the frame-level Content-Encoding, unfortunately the Content-Encoding applies to the entire multi-part stream, if the response is not a single part, and applies to the entire PS3.10 file and not just the dataset after the PS3.10 meta information, so it is not possible to simply apply a zlib container, so the offending note is clarified accordingly.
3	The frame deflate TS is added for Video to PS3.16 Table of Media Types and Transfer Syntax UIDs for Compressed Data in Bulkdata, since it has been added to the Transfer Syntax UIDs for application/dicom Media Types for Video, and these should be consistent.
4	There is not yet a standard Content Type for deflated (RFC 1951) bit streams, so a request has been submitted to IANA for "application/deflate". This is preferred over the alternative of specifying that gzip to be used as a container format, instead of deflate without a gzip; the existing "application/gzip" could be used but unfortunately gzip allows for other compression schemes than deflate to be used, which means the recipient may not have the appropriate decoder. This is the rationale used for the new media type application to IANA for deflate. See also RFC 6713 and RFC 1952 and RFC 1950. Using gzip rather than deflate would also be inconsistent with the existing (entire dataset) Deflated Explicit VR Little Endian Transfer Syntax.

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 that have been segmented, or multi-organ segmentations of large volumetric CT or MR datasets).

Currently, the DICOM standard supports a means of single bit representation of binary segmentations with a Bits Stored and Bits Allocated of 1, and these can grow extremely large, especially when segmenting at the full resolution of the underlying image (e.g., for whole slide imaging). If compressed, they need to be mathematically reversibly (losslessly) compressed. The existing Deflate Transfer Syntax (algorithm used in zip and gzip) is reasonably effective, but applies to the entire data set (including the "metadata" and all the frames treated as a single stream).

Frame-based pixel data compression schemes currently in the standard generally do not support single-bit, with the exception of RLE and J2K (CP 2301), neither of which achieves as high a compression ratio as Deflate does for segmentation data.

Other alternative lossless compression codecs designed for single bit use (such as for fax using CCITT Group 4 (ITU-T T.6), JBIG, or JBIG2) were considered, which though they compress more effectively, were not considered widely enough supported to justify the complexity for this use case at this time. Other general purpose compressors do slightly better than Deflate, but again, not so much better that they justify their addition to the standard at this time, though they may be considered in future if other use cases justify them.

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).

...

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 Explicit VR Little Endian Transfer Syntax defined in Section A.5, which compresses the entire Data Set as a single compressed bit stream.**

Since the Deflate algorithm compresses byte streams regardless of their content, 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.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 Context(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.
       ...

       - ...

       - 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. ...

- ...
- ...

...

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 Explicit VR 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 Section A.2.

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 Deflated Explicit VR Little Endian 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 **"**~~**d**~~**D**eflate**"** 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 **"**~~**d**~~**D**eflate**"** 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**~~**Deflated Explicit VR Little Endian** 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 Compression**	**DeflatedImageFrame Compression**	**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):*

2 Normative References

2.1 International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC)

[ISO/IEC 10918-1] ISO/IEC. 1994. *JPEG Standard for digital compression and encoding of continuous-tone still images. Part 1 - Requirements and implementation guidelines*.

2.2 Internet Engineering Task Force (IETF) and Internet Assigned Names Authority (IANA)

[RFC7230] IETF. June 2014. *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*. http://tools.ietf.org/html/rfc7230 .

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.

...

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 Compression**	**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
	...		
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 Compression**	**O**
	1.2.840.10008.1.2.4.90	JPEG 2000 Image Compression (Lossless Only)	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 Compression**	**O**
	1.2.840.10008.1.2.4.100	MPEG2 Main Profile @ Main Level	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. 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/deflate	**1.2.840.10008.1.2.1.uu**	**Deflated Image Frame Compression**	**D**
	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
		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/deflate	**1.2.840.10008.1.2.1.uu**	**Deflated Image Frame Compression**	**D**
	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

Resource Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	Optionality
		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
		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
	application/deflate	**1.2.840.10008.1.2.1.uu**	**Deflated Image Frame Compression**	**D**
Text		N/A (no defined compression transfer syntaxes for Text)		
Other		N/A (no defined compression transfer syntaxes for Other)		

...

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. A**~~lternatively, if the user agent allowed a~~** Content-Encoding header field of 'deflate'**~~, then~~ cannot be used to transfer** the deflated bytes**~~ may be transferred~~** unaltered**~~, but the Transfer Syntax parameter in the response should be the Explicit VR Little Endian Transfer Syntax~~ since the required PS3.10 File Meta Information is not included in the deflated bytes, and the Content-Encoding applies to the entire multipart stream, not each part within it individually**.

The resource on the origin server may have been encoded in the Deflated Image Frame Compression (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 for a single part response may be transferred after adding a zlib container per [RFC7230] Section 4.2.2, but the Transfer Syntax parameter in the response should be the Explicit VR Little Endian Transfer Syntax.

2. ...