| | |
|---|---|
| Status | Final Text |
| Date of Last Update | 2021/07/05 |
| Person Assigned | David Clunie<br>mailto:dclunie@dclunie.com |
| Submitter Name | IDC<br>mailto:dclunie@dclunie.com |
| Submission Date | 2020/09/28 |

| |
|---|
| Correction Number CP-2083 |
| Log Summary: Add uncompressed encapsulated Transfer Syntax for encoding very large pixel data sets |
| Name of Standard<br>PS3.5, PS3.6 2021b |
| Rationale for Correction:<br>Very large uncompressed Pixel Data may exceed the size limit (2^32-2) of the data element, yet be required for specific applications that need the information in one file but can tolerate fragmentation into frames (e.g., dual-personality DICOM-TIFF WSI files).<br>Add a Transfer Syntax that uses the encapsulation mechanism (sequence items in the Pixel Data element), which permits unlimited length Pixel Data (within the limit of 2^32-1 frames due to the VR of Number of Frames. |
| Correction Wording: |

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

# 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) defined outside the DICOM Standard.

If Pixel Data (7FE0,0010) is sent in a Native Format, then the Photometric Interpretation (0028,0004) shall be other than:

- YBR_RCT
- YBR_ICT
- YBR_PARTIAL_420

**Note**

These values are not permitted because they are not encodable in an uncompressed form.

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.

If Pixel Data (7FE0,0010) is sent in a Native Format, the Value Representation OW is most often required. The Value Representation OB may also be used for Pixel Data (7FE0,0010) in cases where Bits Allocated has a value less than or equal to 8, but only with Transfer Syntaxes where the Value Representation is explicitly conveyed (see Annex A).

**Note**

1. The DICOM default Transfer Syntax (Implicit VR Little Endian) does not explicitly convey Value Representation and therefore the VR of OB may not be used for Pixel Data (7FE0,0010) when using the default Transfer Syntax.
2. The 32-bit Value Length Field limits the maximum size of large data values such as Pixel Data sent in a Native Format.

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.

It is not permitted to have more than one of Pixel Data Provider URL (0028,7FE0), Pixel Data (7FE0,0010), Float Pixel Data (7FE0,0008) or Double Float Pixel Data (7FE0,0009) in the top level Data Set.

**Note**

Pixel Data encoded in Float Pixel Data (7FE0,0008) or Double Float Pixel Data (7FE0,0009) can be considered as consisting of Pixel Cells that entirely occupy the allocated bits, and therefore do not cross word boundaries.

Native format Pixel Cells are encoded as the direct concatenation of the bits of each Pixel Cell, the least significant bit of each Pixel Cell is encoded in the least significant bit of the encoded word or byte, immediately followed by the next most significant bit of each Pixel Cell in the next most significant bit of the encoded word or byte, successively until all bits of the Pixel Cell have been encoded, then immediately followed by the least significant bit of the next Pixel Cell in the next most significant bit of the encoded word or byte. The number of bits of each Pixel Cell is defined by the Bits Allocated (0028,0100) Data Element Value. When a Pixel Cell crosses a word boundary in the OW case, or a byte boundary in the OB case, it shall continue to be encoded, least significant bit to most significant bit, in the next word, or byte, respectively (see ???). For Pixel Data (7FE0,0010) encoded with the Value Representation OW,

the byte ordering of the resulting 2-byte words is defined by the Little Endian Transfer Syntaxes negotiated at the Association Establishment (see Annex A).

**Note**

1. For Pixel Data (7FE0,0010) encoded with the Value Representation OB, the Pixel Data (7FE0,0010) encoding is unaffected by byte ordering.

2. If encoding Pixel Data (7FE0,0010) with a Value for Bits Allocated (0028,0100) not equal to 16 be sure to read and understand ???.

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). The encapsulated pixel stream of encoded pixel data is segmented into one or more Fragments, each of which conveys its own explicit length. The sequence of Fragments of the encapsulated pixel stream is terminated by a delimiter, thus allowing the support of encoding processes where the resulting length of the entire pixel stream is not known until it is entirely encoded. This Encapsulated Format supports both Single-Frame and Multi-Frame images (as defined in PS3.3). At least one frame shall be present, and hence at least one fragment will be present.

**Note**

Depending on the Transfer Syntax, a frame may be entirely contained within a single fragment, or may span multiple fragments to support buffering during compression or to avoid exceeding the maximum size of a fixed length fragment. A recipient can detect fragmentation of frames by comparing the number of fragments (the number of Items minus one for the Basic Offset Table) with the number of frames. Some performance optimizations may be available to a recipient in the absence of fragmentation of frames, but an implementation that fails to support such fragmentation does not conform to the Standard.

...

# 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. The Data Elements contained in the Data Set structure shall be encoded with Explicit VR (with a VR Field) as specified in ???.

2. The encoding of the overall Data Set structure (Data Element Tags, Value Length, etc.) shall be in Little Endian as specified in ???.

3. The encoding of the Data Elements of the Data Set shall be as follows according to their Value Representations:

   - For all Value Representations defined in this Part of the DICOM Standard, except for the Value Representations OB and OW, the encoding shall be in Little Endian as specified in ???.

   - 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 fixed 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 native, it shall have a defined Value Length, and be encoded as follows:

- where Bits Allocated (0028,0100) has a value greater than 8 shall have Value Representation OW and shall be encoded in Little Endian;
- where Bits Allocated (0028,0100) has a value less than or equal to 8 shall have the Value Representation OB or OW and shall be encoded in Little Endian.

**Note**

a. The OL and OV Value Representations are not used for Pixel Data, even if it has a Bits Allocated (0028,0100) of 32 or 64, since OL and OV were added to the Standard after the encoding of Pixel Data had been established

b. That is, as if the Transfer Syntax were Explicit VR Little Endian.

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 Table A.4-1 and Table A.4-2.

- 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 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.

- 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 Table A.4-1).
  - 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 Table A.4-2).

**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 Item Length Field of Value (00000000H) (i.e., no Value Field shall be present).

- Overlay Data (60xx,3000)
  - ...

**Note**

1. For Data encoded with the Value Representation OB, the Data encoding is unaffected by byte ordering.
2. Encoding of Curve Data (5000,3000) and Audio Sample Data (5000,200C) was previously defined but has been retired. See PS3.5-2004.
3. Vertex Point Index List (0066,0025), Edge Point Index List (0066,0024), Triangle Point Index List (0066,0023) and Primitive Point Index List (0066,0029) were previously defined with a value representation of OW and always interpreted as unsigned, but have been retired. These have been replaced by corresponding OL data elements, which allow values larger than 65535 to index the full range of points that can be encoded in Point Coordinates Data (0066,0016). See PS3.5-2015c.

## Table A.4-1. Example for Elements of an Encoded Single-Frame Image Defined as a Sequence of Three Fragments Without Basic Offset Table Item Value

| Pixel Data Element Tag | Value Representation | | Data Element Length | Data Element | | | | |
|---|---|---|---|---|---|---|---|---|
| (7FE0, 0010) with VR of OB | OB | 0000H Reserved | FFFF FFFFH undefined length | **Basic Offset Table with NO Item Value** | | **First Fragment (Single Frame) of Pixel Data** | | |
| | | | | **Item Tag** | **Item Length** | **Item Tag** | **Item Length** | **Item Value** |
| | | | | (FFFE, E000) | 0000 0000H | (FFFE, E000) | 0000 04C6H | Compressed Fragment |
| 4 bytes | 2 bytes | 2 bytes | 4 bytes | 4 bytes | 4 bytes | 4 bytes | 4 bytes | 04C6H bytes |

## Table A.4-1b. Example for Elements of an Encoded Single-Frame Image Defined as a Sequence of Three Fragments Without Basic Offset Table Item Value (continued)

| Data Element Continued | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Second Fragment (Single Frame) of Pixel Data** | | | **Third Fragment (Single Frame) of Pixel Data** | | | **Sequence Delimiter Item** | |
| **Item Tag** | **Item Length** | **Item Value** | **Item Tag** | **Item Length** | **Item Value** | **Sequence Delim. Tag** | **Item Length** |
| (FFFE, E000) | 0000 024AH | Compressed Fragment | (FFFE, E000) | 0000 0628H | Compressed Fragment | (FFFE,E0DD) | 0000 0000H |
| 4 bytes | 4 bytes | 024AH bytes | 4 bytes | 4 bytes | 0628H bytes | 4 bytes | 4 bytes |

## Table A.4-2. Examples of Elements for an Encoded Two-Frame Image Defined as a Sequence of Three Fragments with Basic Table Item Values

| Pixel Data Element Tag | Value Representation | | Data Element Length | Data Element | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| (7FE0, 0010) with VR of OB | OB | 0000H Reserved | FFFF FFFFH undefined length | **Basic Offset Table with Item Value** | | | **First Fragment (Frame 1) of Pixel Data** | | |
| | | | | **Item Tag** | **Item Length** | **Item Value** | **Item Tag** | **Item Length** | **Item Value** |
| | | | | (FFFE, E000) | 0000 0008H | 0000 0000H<br>0000 0646H | (FFFE, E000) | 0000 02C8H | Compressed Fragment |
| 4 bytes | 2 bytes | 2 bytes | 4 bytes | 4 bytes | 4 bytes | 0008H bytes | 4 bytes | 4 bytes | 02C8H bytes |

**Table A.4-2b. Examples of Elements for an Encoded Two-Frame Image Defined as a Sequence of Three Fragments with Basic Table Item Values (continued)**

| Data Element Continued | | | | | | | |
|---|---|---|---|---|---|---|---|
| Second Fragment (Frame 1) of Pixel Data | | | Third Fragment (Frame 2) of Pixel Data | | | Sequence Delimiter Item | |
| Item Tag | Item Length | Item Value | Item Tag | Item Length | Item Value | Sequence Delimiter Tag | Item Length |
| (FFFE, E000) | 0000 036EH | Compressed Fragment | (FFFE, E000) | 0000 0BC8H | Compressed Fragment | (FFFE, E0DD) | 0000 0000H |
| 4 bytes | 4 bytes | 036EH bytes | 4 bytes | 4 bytes | 0BC8H bytes | 4 bytes | 4 bytes |

## A.4.1 JPEG Image Compression

...

## A.4.2 RLE Image Compression

??? defines a RLE Image Compression Transfer Syntax. This transfer Syntax is identified by the UID value "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.4.10 HEVC/H.265 Main 10 Profile / Level 5.1 Video Compression

...

## 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 value of "1.2.840.10008.1.2.1.98".**

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

**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.**

*Add new Transfer Syntax UID to DICOM PS3.6:*

# 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 | Implicit VR Little Endian: Default Transfer Syntax for DICOM | ImplicitVRLittleEndian | Transfer Syntax | PS3.5 |
| 1.2.840.10008.1.2.1 | Explicit VR Little Endian | ExplicitVRLittleEndian | Transfer Syntax | PS3.5 |
| **1.2.840.10008.1.2.1.98** | **Encapsulated Uncompressed Explicit VR Little Endian** | **EncapsulatedUncompressedExplicitVRLittleEndian** | **Transfer Syntax** | PS3.5 |
| 1.2.840.10008.1.2.1.99 | Deflated Explicit VR Little Endian | DeflatedExplicitVRLittleEndian | Transfer Syntax | PS3.5 |

| UID Value | UID Name | UID Keyword | UID Type | Part |
|---|---|---|---|---|
| *1.2.840.10008.1.2.2* | *Explicit VR Big Endian (Retired)* | *ExplicitVRBigEndian* | *Transfer Syntax* | *PS3.5 (2011)* |
| 1.2.840.10008.1.2.4.50 | JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression | JPEGBaseline8Bit | Transfer Syntax | PS3.5 |
| ... | | | | |
| 1.2.840.10008.1.2.5 | RLE Lossless | RLELossless | Transfer Syntax | PS3.5 |
| *1.2.840.10008.1.2.6.1* | *RFC 2557 MIME encapsulation (Retired)* | *RFC2557MIMEEncapsulation* | *Transfer Syntax* | *PS3.10 (2018b)* |
| *1.2.840.10008.1.2.6.2* | *XML Encoding (Retired)* | *XMLEncoding* | *Transfer Syntax* | *PS3.10 (2018b)* |
| 1.2.840.10008.1.2.7.1 | SMPTE ST 2110-20 Uncompressed Progressive Active Video | SMPTEST211020 UncompressedProgressive ActiveVideo | Transfer Syntax | PS3.5 |
| 1.2.840.10008.1.2.7.2 | SMPTE ST 2110-20 Uncompressed Interlaced Active Video | SMPTEST211020 UncompressedInterlaced ActiveVideo | Transfer Syntax | PS3.5 |
| 1.2.840.10008.1.2.7.3 | SMPTE ST 2110-30 PCM Digital Audio | SMPTEST211030PCMDigital Audio | Transfer Syntax | PS3.5 |