

Status	Letter Ballot
Date of Last Update	2023/11/13
Person Assigned	David Clunie mailto:dclunie@dclunie.com
Submitter Name	David Clunie mailto:dclunie@dclunie.com
Submission Date	2023/04/18
Correction Number CP-2302	
Log Summary: Clarify absence of padding between frames applies to single BitsStored multi-frame images encoded in Native format	
Name of Standard	
PS3.5	
Rationale for Correction:	
It is already explicit that there is no padding between frames encoded as Native (not Encapsulated) and that Pixel Cells are packed successively across byte or word boundaries, but it may not be obvious that this means that when BitsAllocated is 1 (e.g., a BINARY Segmentation) that frames may span byte or word boundaries, since there is no provision to allow padding between frames.	
Correction Wording:	

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

## 8 Encoding of Pixel, Overlay and Waveform Data

### 8.1 Pixel and Overlay Data, and Related Data Elements

...

#### 8.1.1 Pixel Data Encoding of Related Data Elements

Encoded Pixel Data of various bit depths shall be accommodated. The following three Data Elements shall define the Pixel structure:

- Bits Allocated (0028,0100)
- Bits Stored (0028,0101)
- High Bit (0028,0102)

Each Pixel Cell shall contain a single Pixel Sample Value. The size of the Pixel Cell shall be specified by Bits Allocated (0028,0100). Bits Stored (0028,0101) defines the total number of these allocated bits that will be used to represent a Pixel Sample Value. Bits Stored (0028,0101) shall never be larger than Bits Allocated (0028,0100). High Bit (0028,0102) specifies where the high order bit of the Bits Stored (0028,0101) is to be placed with respect to the Bits Allocated (0028,0100) specification. Bits Allocated (0028,0100) shall either be 1, or a multiple of 8. High Bit (0028,0102) shall be one less than Bits Stored (0028,0101).

#### Note

1. For example, in Pixel Data with 16 bits (2 bytes) allocated, 12 bits stored, and bit 11 specified as the high bit, one pixel sample is encoded in each 16-bit word, with the 4 most significant bits of each word not containing Pixel Data. See ??? for other examples of the basic encoding schemes.
2. Formerly, bits not used for Pixel Sample Values were described as being usable for overlay planes, but this usage has been retired. See PS3.5-2004.
3. Formerly, High Bit (0028,0102) was not restricted to be one less than Bits Stored (0028,0101) in this Part, or in the general case, though almost all Information Object Definitions in PS3.3 imposed such a restriction. See PS3.5 2014c.
4. Receiving applications may not assume anything about the contents of unused bits, and in particular may not assume that they are zero, or that they contain sign extension bits.

...

Also, the Value Field containing Pixel Data, like all other Value Fields in DICOM, shall be an even number of bytes in length. This means that the Value Field may need to be padded with data that is not part of the image and shall not be considered significant. If needed, the padding bits shall be appended to the end of the Value Field, and shall be used only to extend the data to the next even byte increment of length.

#### Note

The 32-bit Value Length Field limits the maximum size of large data Value Fields such as Pixel Data sent in a Native Format (encoded in Transfer Syntaxes that use only the unencapsulated form).

In a multi-frame object that is transmitted in Native Format, the individual frames are not padded. The individual frames shall be concatenated and padding bits (if necessary) applied to the complete Value Field. At least one frame shall be present.

#### Note

1. Receiving applications should be aware that some older applications may send Pixel Data with excess padding, which was not explicitly prohibited in earlier versions of the Standard. Applications should be prepared to accept such Pixel Data Data Elements, but may delete the excess padding. In no case should a sending application place private data in the padding data.

2. In a multi-frame object with a Bits Allocated (0028,0100) of 1 that is transmitted in Native Format, the individual frames are not padded, therefore successive bits are packed into bytes or words as described in 8.2. I.e., a frame other than the first frame may start in the middle of a byte or word. This is consistent with the historical encoding of Multi-frame Overlays described in 8.1.

## 8.1.2 Overlay Data Encoding of Related Data Elements

Overlay Data is encoded as the direct concatenation of the bits of a single Overlay Plane, where the first bit of an Overlay Plane is encoded in the least significant bit, immediately followed by the next bit of the Overlay Plane in the next most significant bit. For a Multi-frame Overlay, the individual frames are not padded. The individual frames shall be concatenated and padding bits (if necessary) applied to the complete Value Field. When the Overlay Data 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 Overlay Data 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 ???).

## 8.2 Native or Encapsulated Format Encoding

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 ???).

### Note

1. The DICOM Default Little Endian 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 Value Fields such as Pixel Data sent in a Native Format.

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 ???).

### 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 ???.