

1	Status	Final Text
2	Date of Last Update	2016/05/25
3	Person Assigned	David Clunie
4		mailto:dclunie@dclunie.com
5	Submitter Name	David Clunie
6		mailto:dclunie@dclunie.com
7	Submission Date	2015/10/02

8	Correction Number CP-1549
9	Log Summary: Remove description of retired Big Endian Transfer syntax
10	Name of Standard
11	PS3.2, PS3.5 20126b
12	Rationale for Correction:
13	When Supplement 98 flagged the Big Endian Explicit VR Transfer Syntax as retired in PS3.6, it did not remove the description of it
14	in PS3.5.
15	The illustration of the representation inside big and little endian "machines" is not removed.
16	Also clarify that the OL VR is not used for Pixel Data or Overlay Data.
17	Correction Wording:

Amend DICOM PS3.5 as follows:

6.2 Value Representation (VR)

Table 6.2-1. DICOM Value Representations

VR Name	Definition	Character Repertoire	Length of Value
AT Attribute Tag	<p>Ordered pair of 16-bit unsigned integers that is the value of a Data Element Tag.</p> <p>Example: A Data Element Tag of (0018,00FF) would be encoded as a series of 4 bytes in a Little-Endian Transfer Syntax as 18H,00H,FFH,00H and in a Big-Endian Transfer Syntax as 00H,18H,00H,FFH.</p> <p>Note</p> <p>The encoding of an AT value is exactly the same as the encoding of a Data Element Tag as defined in Section 7.</p>	not applicable	4 bytes fixed
OB Other Byte	A string of bytes where the encoding of the contents is specified by the negotiated Transfer Syntax. OB is a VR that is insensitive to Little/Big Endian byte ordering (see Section 7.3). The string of bytes shall be padded with a single trailing NULL byte value (00H) when necessary to achieve even length.	not applicable	see Transfer Syntax definition
OD Other Double	A string of 64-bit IEEE 754:1985 floating point words. OD is a VR that requires byte swapping within each 64-bit word when changing between Little-Endian and Big-Endian byte ordering (see Section 7.3).	not applicable	2^{32} -8 bytes maximum
OF Other Float	A string of 32-bit IEEE 754:1985 floating point words. OF is a VR that requires byte swapping within each 32-bit word when changing between Little-Endian and Big-Endian byte ordering (see Section 7.3).	not applicable	2^{32} -4 bytes maximum
OL Other Long	A string of 32-bit words where the encoding of the contents is specified by the negotiated Transfer Syntax. OL is a VR that requires byte swapping within each word when changing between Little-Endian and Big-Endian byte ordering (see Section 7.3).	not applicable	see Transfer Syntax definition
OW Other Word	A string of 16-bit words where the encoding of the contents is specified by the negotiated Transfer Syntax. OW is a VR that requires byte swapping within each word when changing between Little-Endian and Big-Endian byte ordering (see Section 7.3).	not applicable	see Transfer Syntax definition

6.2.2 Unknown (UN) Value Representation

The Unknown (UN) VR shall only be used for Private Attribute Data Elements and Standard Data Elements previously encoded as some DICOM VR other than UN using the DICOM Default Transfer Syntax (Implicit VR Little Endian), and whose Value Representation is currently unknown. As long as the VR is unknown the Value Field is insensitive to ~~Little/Big Endian~~ byte ordering and shall not be 'byte-swapped' (see Section 7.3). In the case of undefined length sequences, the value shall remain in implicit VR form. See Section 7.8 for a description of Private Data Attribute Elements and section 10 and Annex A for a discussion of Transfer Syntaxes.

Note

- 1.
2. If at some point an application knows the actual VR for an Attribute of VR UN (e.g., has its own applicable data dictionary), it can assume that the Value Field of the Attribute is encoded in Little Endian byte ordering with implicit VR encoding, irrespective of the current Transfer Syntax.
- 3.

1 4.

2 5.

3 **7.3 Big Endian Versus Little Endian Byte Ordering**

4 **All unretired Transfer Syntaxes in DICOM require the use of Little Endian Byte Ordering.**

5 ~~**Another component of the encoding of a Data Set that shall be agreed upon by communicating Application Entities is the**~~
6 ~~**Byte Ordering.**~~

7 Little Endian byte ordering is defined as follows:

- 8 • In a binary number consisting of multiple bytes (e.g., a 32-bit unsigned integer value, the Group Number, the Element Number,
9 etc.), the least significant byte shall be encoded first; with the remaining bytes encoded in increasing order of significance.
- 10 • In a character string consisting of multiple 8-bit single byte codes, the characters will be encoded in the order of occurrence in the
11 string (left to right).

12 Big Endian byte ordering ~~is defined as follows:~~ was previously described but has been retired. See PS3.5 2015x.

- 13 • ~~In a binary number consisting of multiple bytes, the most significant byte shall be encoded first; with the remaining bytes~~
14 ~~encoded in decreasing order of significance.~~
- 15 • ~~In a character string consisting of multiple 8-bit single byte codes, the characters will be encoded in the order of occurrence~~
16 ~~in the string (left to right).~~

17 **Note**

18 The packing of bits within values of OB, ~~OL~~ or OW Value ~~R~~representation for Pixel Data and Overlay Data is described in
19 Section 8. The OL Value Representation is not used for Pixel Data or Overlay Data.

20 Byte ordering is a component of an agreed upon Transfer Syntax (see Section 10). The default DICOM Transfer Syntax, which shall
21 be supported by all AEs, uses Little Endian encoding and is specified in Section A.1. Alternate Little Endian Transfer Syntaxes, ~~some~~
22 ~~of which use Big Endian encoding~~, are also specified in Annex A.

23 **Note**

24 The Command Set structure as specified in PS3.7 is encoded using the Little Endian Implicit VR Transfer Syntax.

25 In the ~~default~~ case of Little Endian encoding, Big Endian Machines interpreting Data Sets shall do 'byte swapping' before interpreting
26 or operating on certain Data Elements. The Data Elements affected are all those having VRs that are multiple byte Values and that
27 are not a character string of 8-bit single byte codes. VRs constructed of a string of characters of 8-bit single byte codes are really
28 constructed of a string of individual bytes, and are therefore not affected by byte ordering. The VRs that are not a string of characters
29 and consist of multiple bytes are:

- 30 • 2-byte US, SS, OW and each component of AT
- 31 • 4-byte OF, OL, UL, SL, and FL
- 32 • 8 byte OD, FD

33 **Note**

34 For the above VRs, the multiple bytes are presented in increasing order of significance when in Little Endian format. For
35 example, an 8-byte Data Element with VR of FD, might be written in hexadecimal as 68AF4B2CH, but encoded in Little
36 Endian would be 2C4BAF68H.

8.1 Pixel and Overlay Data, and Related Data Elements

8.1.2 Overlay Data Encoding of Related Data Elements

...

Note

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 when using the default Transfer Syntax.

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. 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 Annex D). For Overlay Data encoded with the Value Representation OW, the byte ordering of the resulting 2-byte words is defined by the Little Endian ~~or Big Endian~~ Transfer Syntaxes negotiated at the Association Establishment (see Annex A).

Note

For Overlay Data encoded with the Value Representation OB, the Overlay Data encoding is unaffected by ~~Little Endian or Big Endian~~ byte ordering.

8.2 Native or Encapsulated Format Encoding

...

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 Annex D). 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 ~~or Big 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 ~~Little Endian or Big Endian~~ 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 Annex D.

...

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 Element (5400,1010).

Note

Per Section 7.6, an IOD supporting multiple sets of Waveform Data will encapsulate Data Element (5400,1010) within a Sequence.

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

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.

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).

A.2 DICOM Little Endian Transfer Syntax (Explicit VR)

...

Note

1. For Data encoded with the Value Representation OB, the Data encoding is unaffected by ~~Little Endian or Big Endian~~ byte ordering.
2. ...

...

A.3 DICOM Big Endian Transfer Syntax (Explicit VR)

This Transfer Syntax was retired in 2006. For the most recent description of it, see PS3.5 2015X.

~~This Transfer Syntax applies to the encoding of the entire DICOM Data Set. This implies that when a DICOM Data Set is being encoded with the DICOM Big Endian Transfer Syntax the following requirements shall be met:~~

- a. ~~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.~~
- b. ~~The encoding of the overall Data Set structure (Data Element Tags, Value Length, and Value) shall be in Big Endian as specified in Section 7.3.~~
- c. ~~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, except for the Value Representations OB and OW, the encoding shall be in Big Endian as specified in Section 7.3.~~
 - ~~For the Value Representations OB, OL and OW, the encoding shall meet the following specification depending on the Data Element Tag:~~
 - ~~Data Element (7FE0,0010) Pixel Data~~
 - ~~where Bits Allocated (0028,0100) has a value greater than 8 shall have Value Representation OW and shall be encoded in Big 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 Big Endian.~~

Note

~~The OL Value Representation is not used for Pixel Data, even if it has a Bits Allocated (0028,0100) of 32, since OL was added to the standard after the encoding of Pixel Data had been established~~

- ~~Data Element (60xx,3000) Overlay Data~~

- shall have the Value Representation OB or OW and shall be encoded in Big Endian.

Note

~~Previous versions of the Standard specified that the choice of OB or OW VR was based on whether or not Overlay Bits Allocated (60xx,0100) was greater than, or less than or equal to, 8. However, since only one bit plane can be encoded in each Overlay Data Element (60xx,3000), no value of Overlay Bits Allocated other than 1 makes sense. Such a restriction is now present in PS3.3.~~

- ~~Data Element (5400,1010) Waveform Data has the Value Representation specified in its Explicit VR Field. The component points shall be encoded in Big Endian.~~
- ~~Data Elements (0028,1201), (0028,1202), (0028,1203), (0028,1204) Red, Green, Blue, Alpha Palette Lookup Table Data have the Value Representation OW and shall be encoded in Big Endian.~~

Note

~~Previous versions of the Standard either did not specify the encoding of Data Elements (0028,1201), (0028,1202), (0028,1203) in this Part, but specified a VR of US or SS in PS3.6-1993, or specified OW in this Part but a VR of US, SS or OW in PS3.6-1996. The actual encoding of the values and their byte order would be identical in each case, though the explicitly encoded VR field would be different. However, an explicit VR of US or SS cannot be used to encode a table of 2^{16} elements, since the Value Length is restricted to 16 bits.~~

- ~~Data Elements (0028,1101), (0028,1102), (0028,1103) Red, Green, Blue Palette Lookup Table Descriptor have the Value Representation SS or US (depending on rules specified in the IOD in PS3.3), and shall be encoded in Big Endian. The first and third values are always interpreted as unsigned, regardless of the Value Representation.~~
- ~~Data Elements (0028,1221), (0028,1222), (0028,1223) Segmented Red, Green, Blue Palette Color Lookup Table Data have the Value Representation OW and shall be encoded in Big Endian.~~
- ~~Data Element (0028,3006) LUT Data has the Value Representation US or OW and shall be encoded in Big Endian.~~

Note

~~Previous versions of the Standard did not specify the encoding of these Data Elements in this Part, but specified a VR of US or SS in PS3.6-1998. However, an explicit VR of US or SS cannot be used to encode a table of 2^{16} elements, since the Value Length is restricted to 16 bits. Hence a VR of OW has been added. Moreover this element is always unsigned, therefore the VR of SS has been removed. The actual encoding of the values and their byte order would be identical in each case, though the explicitly encoded VR field would be different.~~

- ~~Data Element (0028,3002) LUT Descriptor has the Value Representation SS or US (depending on rules specified in the IOD in PS3.3), and shall be encoded in Big Endian. The first and third values are always interpreted as unsigned, regardless of the Value Representation.~~
- ~~Data Element (0028,1408) Blending Lookup Table Data has the Value Representation OW and shall be encoded in Big Endian.~~
- ~~Data Elements (0066,0025) Vertex Point Index List, (0066,0024) Edge Point Index List, (0066,0023) Triangle Point Index List and (0066,0029) Primitive Point Index List have the Value Representation OW and shall be encoded in Big Endian and are always interpreted as unsigned.~~
- ~~Data Element (0066,0129) Track Point Index List has the Value Representation OL and shall be encoded in Big Endian and is always interpreted as unsigned.~~

Note

1. ~~For Data encoded with the Value Representation OB, the Data encoding is unaffected by Little Endian or Big Endian byte ordering.~~
2. ~~Encoding of Curve Data and Audio Sample Data was previously defined but has been retired. See PS3.5-2004.~~

~~This DICOM Explicit VR Big Endian Transfer Syntax shall be identified by a UID of Value "1.2.840.10008.1.2.2":~~

A.4 Transfer Syntaxes For Encapsulation of Encoded Pixel Data

...

Note

1. For Data encoded with the Value Representation OB, the Data encoding is unaffected by ~~Little Endian or Big Endian~~ byte ordering.
2.

...

D.1 Detailed Example of Pixel Data Encoding

All other (non-default) DICOM Transfer Syntaxes make use of explicit VR encoding. For these Transfer Syntaxes, all Pixel Data where Bits Allocated is less than or equal to 8 may be encoded with an explicit VR of OB (see Annex A). As in the OW case, Pixel Cells are packed together, but in this case the Pixel Data is broken up into a stream of physical 8-bit words.

Note

For Pixel Data encoded with an explicit VR of OB, the encoding of the Pixel Data is unaffected by ~~Little Endian or Big Endian~~ byte ordering.

...

Byte ordering becomes a consideration when we represent the Pixel Data physically, in memory, a file, or on a network.

In the memory of a byte-addressable Big Endian machine, the highest order byte (bits 8 - 15) in each 16-bit word has a binary address of x...x0. While in a byte-addressable Little Endian machine, the lowest order byte (bits 0 - 7) in each 16-bit word has a binary address of x...x0. Figure D-5 pictures our example Pixel Data streams as they would be addressed in the memory of both a Big Endian and a Little Endian machine.

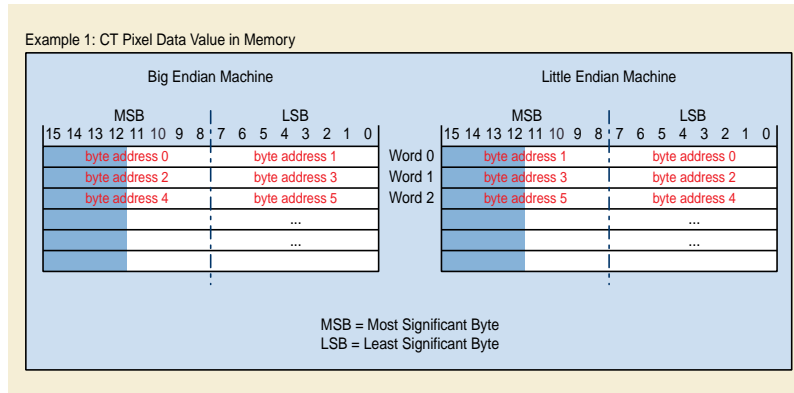


Figure D-5. Example Pixel Cells Byte Ordered in Memory (VR = OW)

Byte ordering is also specified as part of the negotiated Transfer Syntax used in the exchange of a DICOM message. Sixteen bit words are transmitted across the network (a byte at a time) least significant byte first in the case of a Little Endian Transfer Syntax ~~and most significant byte first when using a Big Endian Transfer Syntax~~ (see Figure D-6).

Replace the existing figure:

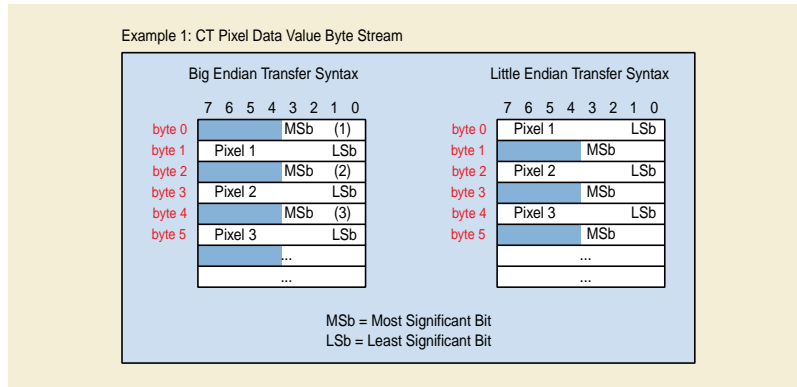


Figure D-6. Sample Pixel Data Byte Streams (VR = OW)

with the new figure:

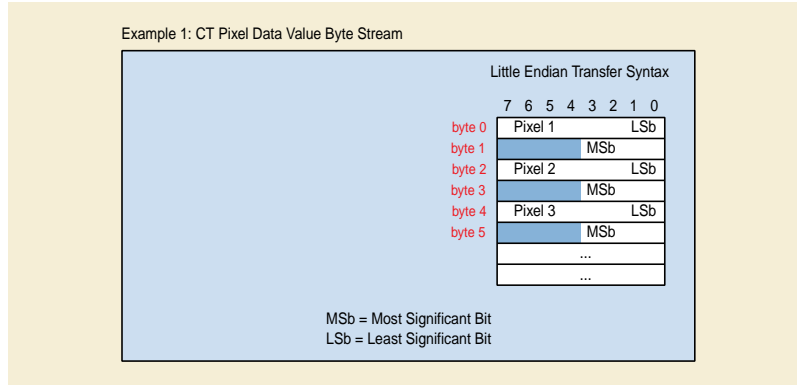


Figure D-6. Sample Pixel Data Byte Streams (VR = OW)

As a last pair of examples, for Pixel Data having the Value Representation OW and the following attributes: 8 bits allocated, 8 bits stored, and a high bit of 7; the resulting byte streams pictured in Figure D-7 are as they would be transmitted across a network and/or stored on media. For Pixel Data having the same attributes, but having the explicit Value Representation OB; the resulting byte streams are unaffected by byte ordering and are pictured in Figure D-8.

Replace the existing figure:

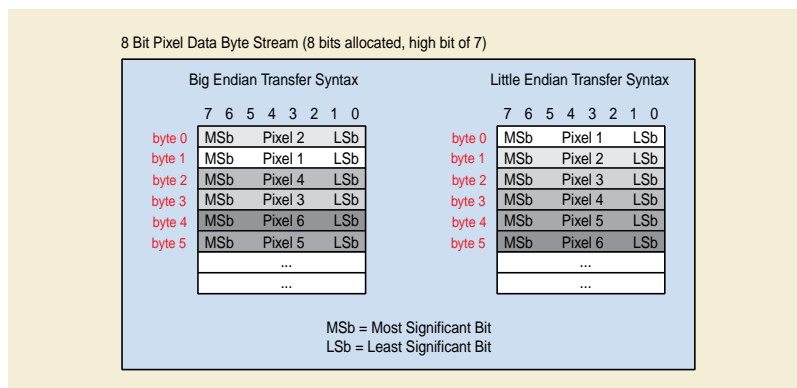


Figure D-7. Sample Pixel Data Byte Streams for 8-bits Allocated and 8-bits Stored (VR = OW)

with the new figure:

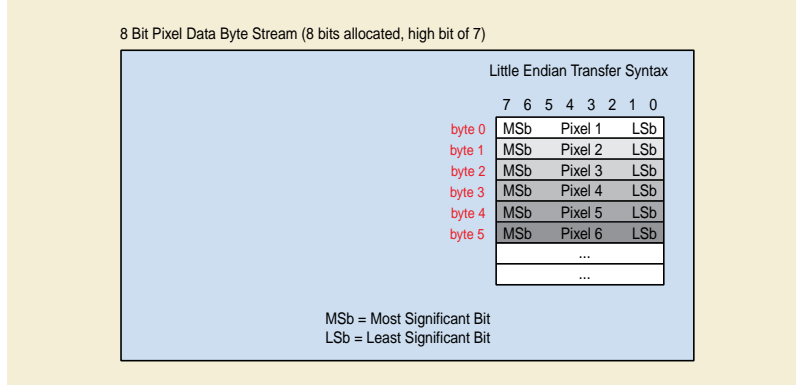


Figure D-7. Sample Pixel Data Byte Streams for 8-bits Allocated and 8-bits Stored (VR = OW)

Replace the existing figure:

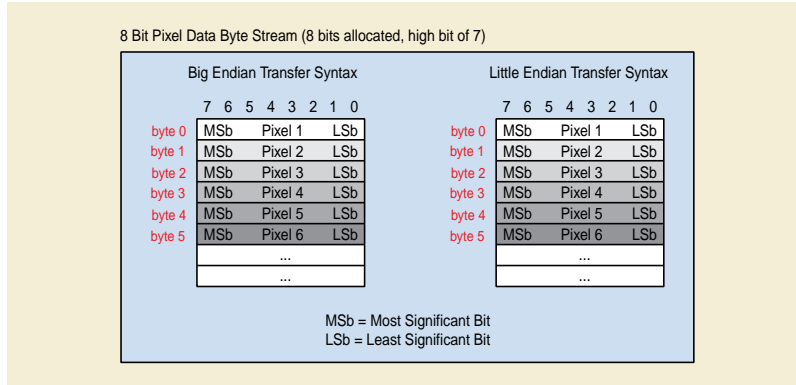


Figure D-8. Sample Pixel Data Byte Streams for 8-bits Allocated and 8-bits Stored (Explicit VR = OB)

with the new figure:

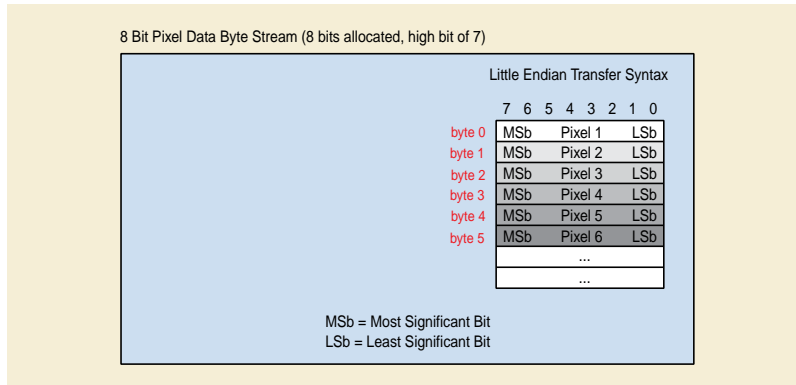


Figure D-8. Sample Pixel Data Byte Streams for 8-bits Allocated and 8-bits Stored (Explicit VR = OB)

D.3 Examples of Float and Double Float Pixel Data

Float Pixel Data having the Value Representation OF always has 32 bits allocated; the resulting byte streams pictured in Figure D.3-1 are as they would be transmitted across a network and/or stored on media.

Replace the existing figure:

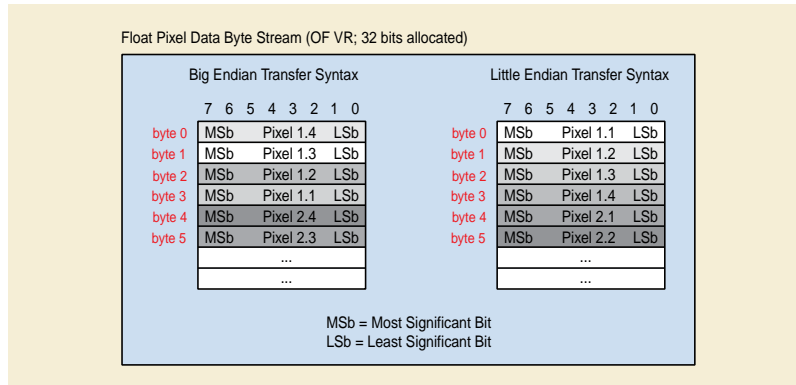


Figure D.3-1. Sample Float Pixel Data Byte Streams for VR = OF

with the new figure:

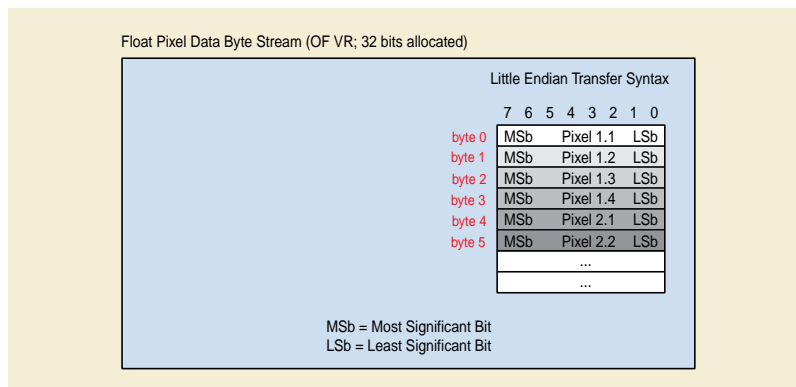
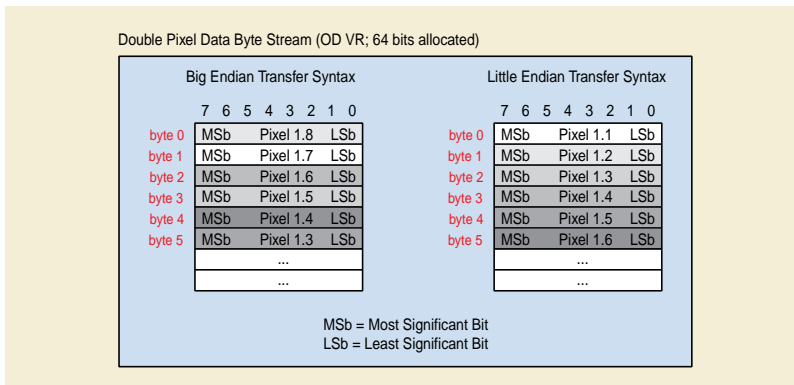


Figure D.3-1. Sample Float Pixel Data Byte Streams for VR = OF

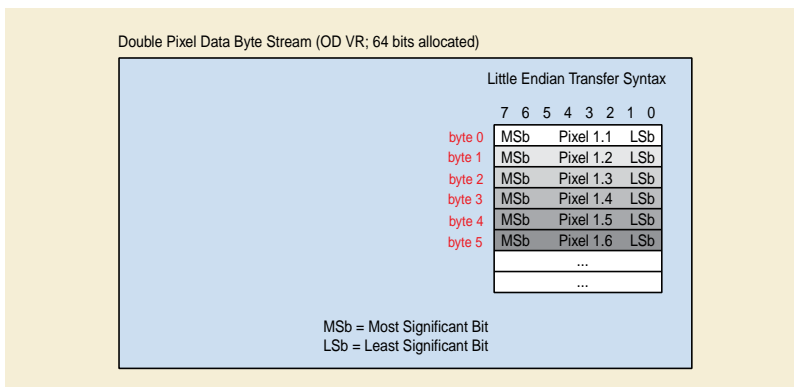
Double Float Pixel Data having the Value Representation OD always has 64 bits allocated; the resulting byte streams pictured in Figure D.3-2 are as they would be transmitted across a network and/or stored on media.

Replace the existing figure:



1 **Figure D.3-2. Sample Float Pixel Data Byte Streams for VR = OD**

2 *with the new figure:*



3 **Figure D.3-2. Sample Float Pixel Data Byte Streams for VR = OD**

4 *Amend DICOM PS3.2 as follows:*

5 **D.4.2.1.4.1.2 Accepted Presentation Contexts**

6 **Table D.4.2-5. Acceptable Presentation Contexts for ECHO-SCP and Receive Echo Request**

7

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
Verification	1.2.840.10008.1.1	Implicit VR Little Endian	1.2.840.10008.1.2	SCP	None
		Explicit VR Little Endian	1.2.840.10008.1.2.1		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

8

9

10

11

12

13

D.4.2.2.4.1.2 Accepted Presentation Contexts

Table D.4.2-10. Acceptable Presentation Contexts for STORAGE-SCP and Receive Storage Request

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
See ???	See ???	Implicit VR Little Endian	1.2.840.10008.1.2	SCP	None
		Explicit VR Little Endian	1.2.840.10008.1.2.1		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

D.4.2.3.3.1.2 Proposed Presentation Contexts

Table D.4.2-16. Proposed Presentation Contexts for STORAGE-SCU and Receive Storage Request

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name List	UID List		
See ???	See ???	Implicit VR Little Endian	1.2.840.10008.1.2	SCU	None
		Explicit VR Little Endian	1.2.840.10008.1.2.1		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		

D.4.2.4.3.1.2 Proposed Presentation Contexts

Table D.4.2-22. Proposed Presentation Contexts for FIND-SCU and Query Remote AE

Presentation Context Table					
Abstract Syntax		Transfer Syntax		Role	Extended Negotiation
Name	UID	Name	UID		
See ???	See ???	Implicit VR Little Endian	1.2.840.10008.1.2	SCU	None
		Explicit VR Little Endian	1.2.840.10008.1.2.1		
		Explicit VR Big Endian	1.2.840.10008.1.2.2		