DICOM Correction Item

Correction Number CP-565	
Log Summary: Pixel Data encoding bit order	
Type of Modification Clarification	Name of Standard PS 3.5 2004
Rationale for Correction Though the encoding of Pixel Data is clear from illustrations in the informative Annex D of PS 3.5, there is no accompanying normative text that actually specifies that pixel data bits are packed from least to most significant bits in the target word or byte. The description for Overlay Data is fairly clear in this respect so the form is reused and generalized to more than one bit to describe Pixel Data encoding. There is also a minor typographic error to be fixed in the Overlay Data section.	
Sections of documents affected PS 3.5 Section 8.2	
Correction Wording:	

Amend PS 3.5:

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. 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 ~~**Pixel**~~ **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).

…

8.2 NATIVE OR ENCAPSULATED FORMAT ENCODING

Pixel data conveyed in the Pixel Data Element (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 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 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: 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.

Native format Pixel Cells are encoded as the direct concatenation of the bits of each Pixel Cell, where ~~**the most significant bit of a Pixel Cell is immediately followed by the least significant bit of the next 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 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).

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

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

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 in one or more Fragments which convey their 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 PS 3.3).