

1	Status	Letter Ballot
2	Date of Last Update	2018/11/11
3	Person Assigned	David Clunie
4		mailto:dclunie@dclunie.com
5	Submitter Name	Rick Busbridge
6		mailto:Rick.Busbridge@philips.com
7	Submission Date	2018/04/05

8	Correction Number CP-1818	
9	Log Summary: Large compressed images may have more frames than fit in the Basic Offset Table	
10	Name of Standard	
11	PS3.3, PS3.5	

1 Rationale for Correction:

2 The Basic Offset Table (first item of encapsulated Pixel Data) uses 32 bit unsigned offsets and these can be exceeded with a large
3 image.

4 It is optional, so does not prevent encoding such an image, but it is more useful as the image grows larger, since otherwise something
5 similar needs to be recalculated by the recipient to get access to random frames.

6 The problem is particularly severe for Whole Slide Microscopy images, which are relatively large (i.e., can exceed 4GB) and for
7 which random frame access is the norm.

8 The work around of using Concatenations is not always desirable.

9 Potential solutions included:

- 10 • Using the same Transfer Syntaxes but doubling the length of the Basic Offset Table since knowing the number of frames a recipient
11 can determine, if the table contains 32-bit or 64-bit pointers. But how would existing recipients know to check for this - do they
12 check the length of the Item for sanity, and if not sane do they reject the image or ignore the offset table? This might break existing
13 implementations.
- 14 • Using the same Transfer Syntaxes but using a special value in the first and second Basic Offset Table that would be invalid or
15 unrealistic as a flag to indicate that the Basic Offset Table uses 64 bit entries subsequently. e.g, a pair of 0xFFFFFFFF entries as
16 a flag. But how would existing recipients know to check for this - do they check the first value or the length of the Item for sanity,
17 and if not sane do they reject the image or ignore the offset table? This might break existing implementations.
- 18 • Adding new Transfer Syntaxes that define different encapsulation rules, e.g., a Basic Offset Table with 64 bit entries, or chained
19 Offset Tables with 32 bit entries that were interspersed in the middle of the sequence of compressed fragments. New Transfer
20 Syntaxes might also consider fixing the number of fragments per frame to 1, since fragmentation is probably not useful in these
21 use cases. This approach would mean that recipients would have to add support for the new Transfer Syntaxes if they want to
22 assure receiving offset tables (assuming the sender bothered to create them).
- 23 • Adding a new Attribute to the top level data set that contains the offset information and requiring that the BOT in the first item of
24 Pixel Data be empty; this would require addition of a new VR for 64 bit unsigned integers; it also raises the question of invalidation
25 of the offsets in the Attribute during transfer/copying of the files that involves Transfer Syntax changes or reorganization of the
26 fragmented frames, as well as whether or not to remove the (irrelevant and relatively large) attribute when retrieving metadata
27 only (e.g., with WADO-RS). This approach would not break existing implementations existing systems, but would mean that
28 recipients would have to add support for the new Attribute if they want to assure receiving offset tables (assuming the sender
29 bothered to create them).
- 30 • Requiring support for Concatenations, since each Instance of a Concatenation can be limited in size so as not to exceed 32 bit
31 offsets to frames. This approach would mean that recipients would have to add support for Concatenations if they want to assure
32 receiving offset tables (assuming the sender bothered to create them).

33 The approach of adding a new Attribute is proposed, which depends on a separate CP 1819 that adds an OV (64 bit unsigned) VR.

34 The length of each encoded frame is also included, since having the information centralized considerably simplifies random access
35 to frames without having to reindex the data when there is a single fragment per frame.

36 Correction Wording:

Amend DICOM PS3.3 as follows (changes to existing text are bold and underlined for additions and ~~struckthrough~~ for removals):

3 Definitions

For the purposes of this Standard the following definitions apply.

3.5 DICOM Data Structures and Encoding

This Part of the Standard makes use of the following terms defined in ????:

Basic Offset Table See PS3.5.

Extended Offset Table **See PS3.5.**

4 Symbols and Abbreviations

The following symbols and abbreviations are used in this Part of the Standard.

BOT **Basic Offset Table**

EOT **Extended Offset Table**

C.7.6.3 Image Pixel Module

Table C.7-11a describes the Image Pixel Module.

Table C.7-11a. Image Pixel Module Attributes

Attribute Name	Tag	Type	Attribute Description
<i>Include Table C.7-11c "Image Pixel Description Macro Attributes"</i>			Required if the IOD is not being transferred in a STOW-RS Request and is not encoded as metadata and compressed bulk pixel data. May be present otherwise. See PS3.18 Section 6.6 "STOW-RS Request/Response". Note When the IOD is encoded as metadata in a STOW-RS Request and the bulk pixel data is compressed, the STOW-RS origin server is required to be able to derive appropriate values for the Image Pixel Macro Attributes from the compressed bit stream.
Pixel Data	(7FE0,0010)	1C	A data stream of the pixel samples that comprise the Image. See Section C.7.6.3.1.4 for further explanation. Required if Pixel Data Provider URL (0028,7FE0) is not present.
Pixel Data Provider URL	(0028,7FE0)	1C	A URL of a provider service that supplies the pixel data of the Image. Required if the image is to be transferred in one of the following presentation contexts identified by Transfer Syntax UID: <ul style="list-style-type: none"> 1.2.840.10008.1.2.4.94 (DICOM JPIP Referenced Transfer Syntax) 1.2.840.10008.1.2.4.95 (DICOM JPIP Referenced Deflate Transfer Syntax) Note The VR of this Data Element has changed from UT to UR.
...

Attribute Name	Tag	Type	Attribute Description
<u>Extended Offset Table</u>	<u>(0028,eee1)</u>	<u>3</u>	<p><u>Byte offsets of the Frames in the Sequence of Items in Encapsulated Pixel Data encoded in Pixel Data (7FE0,0010). See Section C.7.6.3.1.X.</u></p> <p><u>May only be present when:</u></p> <ul style="list-style-type: none"> <u>Pixel Data (7FE0,0010) is present, and</u> <u>the Transfer Syntax uses Encapsulated Pixel Data, and</u> <u>the Transfer Syntax encodes Frames in separate Fragments.</u> <u>the Basic Offset Table in the first Item is not present (its Item has zero length), and</u> <u>each Frame is entirely contained within one Fragment.</u> <p>Note</p> <p><u>Unlike a Basic Offset Table, an Extended Offset Table is not permitted to be empty.</u></p>
<u>Extended Offset Table Lengths</u>	<u>(0028,eee2)</u>	<u>1C</u>	<p><u>Byte lengths of the Frames in the Sequence of Items in Encapsulated Pixel Data encoded in Pixel Data (7FE0,0010). See Section C.7.6.3.1.X.</u></p> <p><u>Required if Extended Offset Table (0028,eee1) is present and every frame is encoded as a single fragment.</u></p> <p>Note</p> <p><u>The length information is not sent when there are multiple fragments per frame, since it is necessary to assemble the frame from fragments with delimiters rather than assuming a single contiguous span of bytes.</u></p>

C.7.6.3.1 Image Pixel Attribute Descriptions

...

C.7.6.3.1.X Extended Offset Table

The Extended Offset Table (0028,eee1) Value shall contain byte offsets to the first byte of the Item Tag of the first fragment for every frame in the Pixel Data Sequence.

The byte offsets are measured from the first byte of the first Item Tag following the empty (zero length) Basic Offset Table item, i.e., the Item Tag of the first fragment of the first frame.

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

3 Definitions

For the purposes of this standard, the following definitions apply.

3.10 DICOM Data Structures and Encoding Definitions

The following definitions are commonly used in this Standard:

Basic Offset Table: A table of 32-bit pointers to individual frames of an encapsulated multi-frame image.

Extended Offset Table: A table of 64-bit pointers to individual frames of an encapsulated multi-frame image.

4 Symbols and Abbreviations

The following symbols and abbreviations are used in this Standard.

BOT Basic Offset Table

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 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 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

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.

- ...
- 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.

- ...
- 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 (0028.eee1) is present.**

• ...

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					
	OB	0000H Reserved		Basic Offset Table with Item Value			First Fragment (Frame 1) of Pixel Data		
(7FE0, 0010) with VR of OB			FFFF FFFFH undefined length	Item Tag	Item Length	Item Value	Item Tag	Item Length	Item Value
				(FFFE, E000)	0000 0008H	0000 0000H 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

Amend DICOM PS3.6 as follows (changes to existing text are bold and underlined for additions and ~~struckthrough~~ for removals):

6 Registry of DICOM Data Elements

Table 6-1. Registry of DICOM Data Elements

Tag	Name	Keyword	VR	VM	
<u>(0028.eee1)</u>	<u>Extended Offset Table</u>	<u>ExtendedOffsetTable</u>	<u>OV</u>	<u>1</u>	
<u>(0028.eee2)</u>	<u>Extended Offset Table Lengths</u>	<u>ExtendedOffsetTableLengths</u>	<u>OV</u>	<u>1</u>	