
1	Status	Draft Final Text
2	Date of Last Update	2024/11/02
3	Person Assigned	David Clunie
4		mailto:dclunie@dclunie.com
5	Submitter Name	David Clunie
6		mailto:dclunie@dclunie.com
7	Submission Date	2024/02/23
8	Correction Number CP-2388	
9	Log Summary: Extended Offset Table clarification	
10	Name of Standard	
11	PS3.3, PS3.5 2024d	

1	Rationale for Correction:
2	The Extended Offset Table introduced by CP 1818 contains offsets and lengths and requires each frame be encoded in a single
3	fragment.
4	Yet the description of the length attribute implies that the offsets may be present when frames are fragmented. Clarify that this is not
5	the case (since the "and every frame is encoded as a single fragment" conditional clause is always true).
6	Also, note that lengths may be odd if the compressed bitstream is an odd length (i.e., padding is not included).
7	Also, clarify the description of when the Basic Offset Table is not present (i.e., the Item Tag is always present, but when it is zero
8	value length, the table is absent).
9	Also, explain why offset table is only present for Encapsulated Format.
10	Not relaxing the requirement that the BOT and EOT are mutually exclusive, since hard to find a use case for both, and they are
11	bulky, even though legacy readers might use the BOT, and new readers might take advantage of the EOT being in the metadata,
12	or having the additional length information.
13	Not adding another optional Attribute that is the byte offset from the start of the PS3.10 file to the beginning of the Pixel Data 7FE0
14	group, even though this would allow readers to skip to the extended offset table and then the pixel data without fully parsing the
15	header before that point, especially if assigned a low group,element tag number, it would be encoded early in the stream, e.g., after
16	the retired (0008,0001) Length to End (ACR-NEMA message length); alternatively, it could be in the PS3.10 meta information (group
17	0x0002). This could get stale if header is modified;, though the same already applies to EOT.
18	<i>[Comment, UG Austria: Extended Offset Table is only used in case of each frame being encoded into a single fragment (according</i>
19	<i>conditions in Line 11 and 14, also line 27 ff) So the wording in line 35 and line 38 is not exact, as there is a "... first fragment for</i>
20	<i>every frame ..." or "... first fragment of the first frame". FIXED.]</i>
21	<i>[Comment. Carl Zeiss: page 3 line 1: 00000000H should be 0000000000000000H to match VR=OV. FIXED. :)]</i>
22	<i>[Comment. Carl Zeiss: page 1 line 25: missing 'i' in "...without fully parsng...". FIXED.]</i>
23	<i>[Comment. GE: Rationale:"This could get stale if header is modified;, though the same already applies to EOT":Instead of attribute</i>
24	<i>at start of file, one could be placed at the end of the file. It is much more unlikely this would get stale.A warning note to implementors</i>
25	<i>that EOT cannot be taken as gospel might be good to have. Eg, a system not supporting EOT might recompress the pixel data but</i>
26	<i>not update the EOT. FIXED. Warning added in note.]</i>
27	<i>[Comment, GE: C.7.6.3.1.8:"...first fragment for every frame in": This has become superfluous I would think. FIXED.]</i>
28	<i>[Comment, GE: C.7.6.3.1.8: "The byte offsets are measured from the first byte of the first Item Tag following the empty (zero length)</i>
29	<i>Basic Offset Table item, i.e., the Item Tag of the first fragment of the first frame."Again, first fragment reference can be removed.</i>
30	<i>FIXED.]</i>
31	<i>[Comment, GE: C.7.6.3.1.8: Note: Reference to "first byte of Item Tag" is confusing as it is not clear if this is the tag itself or its</i>
32	<i>content. FIXED. Because Item Tag is capitalized, it should be clear that (just like in the PS3.5 description of the BOT), the first byte</i>
33	<i>being referred to is that of the Item Tag, not the Value field of the Item. However, a note and a reference to the illustration in PS3.5</i>
34	<i>is added for clarity. That said, since the Item Tag and Length have a fixed length, it doesn't actually matter, since the offsets are</i>
35	<i>relative.]</i>
36	Correction Wording:

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

C.7.6.3 Image Pixel Module

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

Attribute Name	Tag	Type	Attribute Description
Extended Offset Table	(7FE0,0001)	3	<p>Byte offsets of the <u>Item Tags of the</u> Frames in the Sequence of Items in Encapsulated Pixel Data encoded in Pixel Data (7FE0,0010).</p> <p>See Section C.7.6.3.1.8.</p> <p>May only be present when:</p> <ul style="list-style-type: none"> • Pixel Data (7FE0,0010) is present, and • the Transfer Syntax uses Encapsulated <u>Format for the</u> Pixel Data (<u>7FE0,0010</u>), and • the Transfer Syntax encodes Frames in separate Fragments, and • the Basic Offset Table is not present (i.e., in the first Item <u>of Pixel Data (7FE0,0010) is not present (its Item</u> has zero length), and • each Frame is entirely contained within one Fragment. <p>Note</p> <ol style="list-style-type: none"> 1. Unlike a Basic Offset Table, an Extended Offset Table, if the Attribute is present, is not permitted to be empty. 2. If this Instance is part of a Concatenation, only the offset and lengths of the frames encoded in the Pixel Data (7FE0,0010) of this Instance are indexed in the Extended Offset Table (7FE0,0001) and Extended Offset Table Lengths (7FE0,0002) in this Instance, not those of the entire Concatenation. I.e., the Values of these two Attributes are specific to each Instance. See also Section C.7.6.16.2.2.4. 3. <u>If the length of the Dataset before Pixel Data (7FE0,0010) is modified during transfer or storage, or the Dataset is re-encoded (such as in a different Transfer Syntax) any Extended Offset Table needs to be recomputed or removed.</u>

Attribute Name	Tag	Type	Attribute Description
Extended Offset Table Lengths	(7FE0,0002)	1C	<p>Byte lengths of the Frames in the Sequence of Items in Encapsulated Pixel Data encoded in Pixel Data (7FE0,0010).</p> <p>See Section C.7.6.3.1.8.</p> <p>Required if Extended Offset Table (7FE0,0001) is present and every frame is encoded as a single fragment.</p> <p>Note</p> <ol style="list-style-type: none"> 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. This Attribute is only sent when each Frame is entirely contained within one Fragment as a single contiguous span of bytes so that it is not necessary to assemble the frame from fragments with delimiters. <u>The length encoded in this Attribute may be an odd number if the compressed bitstream for the frame is an odd length; i.e., it does not include any trailing padding required to make the Item an even length.</u>

C.7.6.3.1 Image Pixel Module Attribute Descriptions

C.7.6.3.1.8 Extended Offset Table

The Extended Offset Table (7FE0,0001) Value shall contain byte offsets to the first byte of the Item Tag of the **firstone and only** fragment for every frame in the Pixel Data (7FE0,0010) 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 **firstone and only** fragment of the first frame.

Note

The offset is to the first byte of the Item Tag itself, not the first byte of the Value field within the Item. See also the example of the Basic Offset Table in PS3.5 Table A.4-2.

If present, the first entry will always be 0000000000000000H.

Note

The offset table is only present when the Pixel Data (7FE0,0010) is encoded in Encapsulated Format, since the position of uncompressed frames encoded in Native Format can be computed from their known fixed length (i.e., derived from Rows (0028,0010), Columns (0028,0011), Samples per Pixel (0028,0002) and Bits Allocated (0028,0100)).

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

A.4 Transfer Syntaxes For Encapsulation of Encoded Pixel Data

-
-
- 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, 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).

...

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 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

- 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.
- 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.
- 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.
- If the Basic Offset Table Item Value is not zero length, the first entry will always be 00000000H.**
- The offset is to the first byte of the Item Tag itself, not the first byte of the Value field within the Item**

- ...

Note

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
				4 bytes	2 bytes	2 bytes	4 bytes	(FFFE, E000)	0000 0008H
				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

1

2

3

4

5

6

Data Element Continued							
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