

### DICOM Correction Proposal Form

Tracking Information - Administrator Use Only	
Correction Proposal Number	CP -123
STATUS	Assigned - revision <del>0203</del>
Date of Last Update	<del>98/01/31</del> 1999/10/20
Person Assigned	David Clunie <a href="mailto:dclunie@idt.net">dclunie@idt.net</a>
Log Summary: Clarification of Unknown VR byte order issues with new UB VR	
Submitter Name: Herve Hoehn	Submission Date: 98/01/06
Type of Modification: Clarification	
Name of Document: PS 3.5 <del>1996-1999</del> Data Structure and Encoding <del>as amended by Supplement 14</del>	Version Number:
<p>Rationale for Change:</p> <p>Supplement 14 adds a VR of UN (Unknown) to allow attributes received in default Implicit VR Little Endian Transfer Syntax to be preserved in explicit VR even if the dictionary VR is not known (eg. for new standard attributes or private attributes).</p> <p>The byte order for UN is defined to be Little Endian (in case the "real" VR is multibyte).</p> <p>The same supplement also creates the conditions for the extension of the standard with new VRs by defining the encoding form of the Value Length and VR fields.</p> <p>This creates the possibility that an implementation will receive an attribute in an Explicit VR Transfer Syntax, but with a VR that it does not recognize.</p> <p>In this case, if the Transfer Syntax is Little Endian, then the attribute may safely be stored and regurgitated as UN.</p> <p>On the other hand, if the attribute is received in an Explicit Big Endian Transfer Syntax, then the implementation that does not recognize the VR does not know whether to store or transmit the value in the same or swapped byte order, since the unrecognized VR may be single or multi-byte.</p> <p>This correction adds another VR, UB, to specifically handle the Big Endian case, as well as describing the use of UN in the Little Endian case.</p> <p>This correction still does not deal with the case when an unrecognized VR is received in Explicit VR Big Endian Transfer Syntax that needs to be retransmitted in Implicit VR Little Endian.</p> <p><u>An alternative proposal solution to that described here is to require all new VRs to make use of the spare padding bytes in the "long form" of explicit VR encoding to encode the actual length in bytes</u></p>	

the explicit transfer syntax in which the data was received.

This second proposal does handle the case when an unrecognized VR is received in Explicit VR Big Endian Transfer Syntax that needs to be retransmitted in Implicit VR Little Endian.

Sections of Document Affected:

Section 6.2, 6.4, 7.1.2

Suggested Wording of Changes:

### Alternative proposal 1:

Item #1

Change section 6.2 - all modifications to the existing section are shown in **BOLD underlined** font.

All new VRs defined in future versions of DICOM shall be of the same Data Element Structure as defined in Section 7.1.2 (i.e. following the format for VRs such as OB, OW, SQ, **UB** and UN).

Notes: **1.** Since all new VRs will be defined as specified in section 7.1.2, an implementation may choose to ignore VRs not recognized by applying the rules stated in Section 7.1.2.

**2. If an Attribute with an unrecognized VR needs to be stored or retransmitted, the UN or UB VR may be used, depending on the byte order of the Transfer Syntax in which it was received.**

Item #2

Amend UN VR section 6.2.1 to include UB

#### **6.2.1 Unknown (UN and UB) Value Representations**

The Unknown (UN) VR shall only be used for Private Attribute Data Elements and Standard Data Elements previously encoded as:

**1.** some DICOM VR other than UN using the DICOM Default Transfer Syntax (Implicit VR Little Endian), and whose Value Representation is currently unknown- , **or**

**2. some new DICOM VR not recognized by the implementation using any DICOM Little Endian Transfer Syntax.**

**The Unknown Big Endian (UB) VR shall only be used for Private Attribute Data Elements and Standard Data Elements previously encoded as some new DICOM VR not recognized by the implementation using any DICOM Big Endian Transfer Syntax.**

As long as the VR is **unknown UN or UB** the Value Field is insensitive to Little/Big Endian byte ordering and shall not be 'byte-swapped' (see section 7.3). See section 7.8 for a description of Private Data Attribute Elements and section 10 and Annex A for a discussion of Transfer Syntaxes.

The UN VR shall not be used for Private Creator Data Elements (i.e. the VR is equal to LO, see section 7.8.1).

Notes: **1.** All other (non-default) DICOM Transfer Syntaxes employ explicit VR in their encoding, and therefore any Private and/or Standard Data Element Value Field Attribute value encoded and decoded using any Transfer Syntax other than the default, and not having been translated to the DICOM Default Transfer Syntax default in the interim, will have a known VR, **though that VR may not be known to all implementations.**

**2.** If at some point an application knows the actual VR for an Attribute of VR UN **or** UB (e.g. has its own

3. ~~This~~ **These** VRs of UN ~~and UB is~~ **are** needed when an explicit VR must be given to a Data Element whose Value Representation is unknown (e.g. store and forward). UN ~~and UB is a~~ **are** means to explicitly indicate that the Value Representation of a Data Element is unknown.

4. The length field of the Value Representation of UN **or UB** may contain the value of “unknown length”. See section 7.5.1 to determine how to parse Data Elements with an unknown length.

5. An example of a Standard Data Element using a UN VR is a Type 3 or Type U Standard Attribute added to an SOP Class definition. An existing application which does not support that new Attribute (and encounters it) could convert the VR to UN.

**6. An example of a Standard Data Element using a UB VR is a Type 3 or Type U Standard Attribute added to an SOP Class definition that uses a new VR added to PS 3.5. An existing application which does not support that new Attribute (and encounters it in an Explicit Big Endian Transfer Syntax) could convert the VR to UB.**

Item #3

Amend UN and add UB VR into Table 6.2-1

Table 6.2-1: DICOM value representations

VR Name	Definition	Character Repertoire	Length of Value
<b><u>UB</u></b> <b><u>Unknown Big Endian</u></b>	<b><u>A string of bytes where the encoding of the contents is unknown, but multibyte values are big endian. (See Section 6.2.1)</u></b>	<b><u>not applicable</u></b>	<b><u>Any length valid for any of the other DICOM Value Representations</u></b>
UN Unknown <b><u>Little Endian</u></b>	A string of bytes where the encoding of the contents is unknown, <b><u>but multibyte values are little endian.</u></b> (See Section 6.2.1)	not applicable	Any length valid for any of the other DICOM Value Representations

Item #4

Change section 6.4 to include UB VR - all modifications to the existing section are shown in **BOLD underlined** font.

Data Elements with a VR of SQ, OW, OB, **UB** or UN shall always have a Value Multiplicity of one.

Item #5

Change section 7.1.2 to include UB VR - all modifications to the existing section are shown in **BOLD underlined** font.

When using the Explicit VR structures, the Data Element shall be constructed of four consecutive fields: Data Element Tag, VR, Value Length, and Value. Depending on the VR of the Data Element, the Data Element will be structured in one of two ways:

- for VRs of OB, OW, SQ, **UB** and UN the 16 bits following the two character VR Field are reserved for use by later versions of the DICOM Standard. These reserved bytes shall be set to 0000H and shall not be used or decoded (Figure 7.1.2-1). The Value Length Field is a 32-bit unsigned integer. If the Value Field has an Explicit Length, then the Value Length Field shall contain a value equal to the length (in bytes) of the Value Field. Otherwise, the Value Field has an Undefined Length and a Sequence Delimitation Item marks the end of the Value Field.
- for all other VRs the Value Length Field is the 16-bit unsigned integer following the two

Item #6

Add UN to Figures 7.1.2-1 and 7.1.2-2 - all modifications to the existing section are shown in **BOLD Underline font**.

**Figure 7.1.2-1: Data Element with Explicit VR of OB, OW, SQ, UB or UN**

Tag		VR		Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 byte character string) of "OB", "OW", "SQ", " <b><u>UB</u></b> " or "UN"	Reserved (2 bytes) set to a value of 0000H	32-bit unsigned integer	Even number of bytes containing the Data Element Value(s) encoded according to the VR and negotiated Transfer Syntax. Delimited with Sequence Delimitation Item if of Undefined Length.
2 bytes	2 bytes	2 bytes	2 bytes	4 bytes	'Value Length' bytes if of Explicit Length

**Figure 7.1.2-2: Data Element with Explicit VR other than OB, OW, SQ, UB or UN**

Tag		VR	Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 byte character string)	Length (16-bit unsigned integer)	Even number of bytes containing the Data Element Value(s) encoded according to the VR and negotiated Transfer Syntax.
2 bytes	2 bytes	2 bytes	2 bytes	'Value Length' bytes

**Alternative proposal 2:**

**Item #1**

**Change section 7.1.2 to describe the use of the VR Length field - all modifications to the existing section are shown in BOLD underlined font.**

**When using the Explicit VR structures, the Data Element shall be constructed of four consecutive fields: Data Element Tag, VR, Value Length, and Value. Depending on the VR of the Data Element, the Data Element will be structured in one of two ways:**

- **for VRs of OB, OW, SQ and UN the 16 bits following the two character VR Field are used as follows:**
  - **the first byte may explicitly specify the length of a single value encoded in that VR, i.e. 1, 2 or 4 bytes**
- Note:**
- 1. In previous versions of the standard, this byte was reserved and set to 00H.**
  - 2. If this field contains zero, then the receiver must use its implicit knowledge of the characteristics of the VR to determine the actual length of a single value.**
  - 3. The VR length can be used by receivers that encounter unexpected VRs (e.g.**

**4. It is recommended that all VRs added to the standard in future make the use of this length field mandatory.**

- **the second byte is reserved for use by later versions of the DICOM Standard. These This reserved bytes shall be set to 0000H and shall not be used or decoded (Figure 7.1.2-1).**
- **The Value Length Field is a 32-bit unsigned integer. If the Value Field has an Explicit Length, then the Value Length Field shall contain a value equal to the length (in bytes) of the Value Field. Otherwise, the Value Field has an Undefined Length and a Sequence Delimitation Item marks the end of the Value Field.**
- **for all other VRs the Value Length Field is the 16-bit unsigned integer following the two character VR Field (Figure 7.1.2-2). The value of the Value Length Field shall equal the length of the Value Field.**

Item #2

Add use of VR Length field in reserved bytes to Figures 7.1.2-1- all modifications to the existing section are shown in BOLD Underline font.

**Figure 7.1.2-1: Data Element with Explicit VR of OB, OW, SQ, UB or UN**

<b><u>Tag</u></b>		<b><u>VR</u></b>			<b><u>Value Length</u></b>	<b><u>Value</u></b>
<b><u>Group Number (16-bit unsigned integer)</u></b>	<b><u>Element Number (16-bit unsigned integer)</u></b>	<b><u>VR (2 byte character string) of "OB", "OW", "SQ" or "UN"</u></b>	<b><u>Reserved (2 bytes) set to a value of 0000H</u></b> <b><u>VR Length</u></b>	<b><u>Reserved (1 byte) set to a value of 00H</u></b>	<b><u>32-bit unsigned integer</u></b>	<b><u>Even number of bytes contain the Data Element Value(s) encoded according to the VR or negotiated Transfer Syntax. Delimited with Sequence Delimitation Item if of Undefined Length.</u></b>
<b><u>2 bytes</u></b>	<b><u>2 bytes</u></b>	<b><u>2 bytes</u></b>	<b><u>2 1 bytes</u></b>	<b><u>1 byte</u></b>	<b><u>4 bytes</u></b>	<b><u>'Value Length' bytes if of Explicit Length</u></b>