

1	Status	Letter Ballot
2	Date of Last Update	2016/05/25
3	Person Assigned	David Clunie
4		mailto:dclunie@dclunie.com
5	Submitter Name	Mathieu Malaterre
6		mailto:mathieu.malaterre@gmail.com
7	Submission Date	2015/12/18

8	Correction Number CP-1564	
9	Log Summary: Copying unrecognized VRs	
10	Name of Standard	
11	PS3.5	
12	Rationale for Correction:	
13	The standard anticipates an existing implementation encountering unrecognized (new standard) Value Representations and describes how they may be ignored since the parsing rules are defined.	
14		
15	However, the standard is silent on how such data elements with unrecognized VRs should be copied, and though this may be deduced from first principles, it would be helpful to describe it.	
16		
17	Correction Wording:	

Amend DICOM PS3.5 as follows:

6.2 Value Representation (VR)

The Value Representation of a Data Element describes the data type and format of that Data Element's Value(s). ??? lists the VR of each Data Element by Data Element Tag.

Values with VRs constructed of character strings, except in the case of the VR UI, shall be padded with SPACE characters (20H, in the Default Character Repertoire) when necessary to achieve even length. Values with a VR of UI shall be padded with a single trailing NULL (00H) character when necessary to achieve even length. Values with a VR of OB shall be padded with a single trailing NULL byte value (00H) when necessary to achieve even length.

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, OD, OF, OL, OW, SQ and UN).

Note

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. **When converting a Data Set from an Explicit VR Transfer Syntax to a different Transfer Syntax, an implementation may copy Data Elements with unrecognized VRs in the following manner:**
 - **If the endianness of the Transfer Syntaxes is the same, the Value of the Data Element may be copied unchanged and if the target Transfer Syntax is Explicit VR, the VR bytes copied unchanged. In practice this only applies to Little Endian Transfer Syntaxes, since there was only one Big Endian Transfer Syntax defined.**
 - **If the source Transfer Syntax is Little Endian and the target Transfer Syntax is the (retired) Big Endian Explicit VR Transfer Syntax, then the Value of the Data Element may be copied unchanged and the VR changed to UN, since being unrecognized, whether or not byte swapping is required is unknown. If the VR were copied unchanged, the byte order of the value might or might not be incorrect.**
 - **If the source Transfer Syntax is the (retired) Big Endian Explicit VR Transfer Syntax, then the Data Element cannot be copied, because whether or not byte swapping is required is unknown, and there is no equivalent of the UN VR to use when the value is big endian rather than little endian.**

The issues of whether or not the element may be copied, and what VR to use if copying, do not arise when converting a Data Set from Implicit VR Little Endian Transfer Syntax, since the VR would not be present to be unrecognized, and if the data element VR is not known from a data dictionary, then UN would be used.

An individual Value, including padding, shall not exceed the Length of Value...

7.1.2 Data Element Structure with Explicit VR

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, OD, OF, OL, OW, SQ 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 (Table 7.1-1). 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 VRs of UC, UR and UT 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. The Value Length Field is a 32-bit unsigned integer. The Value Field is required to have an Explicit Length, that is the Value Length Field shall contain a value equal to the length (in bytes) of the Value Field.

Note

VRs of UC, UR and UT may not have an Undefined Length, i.e., a Value Length of FFFFFFFFH.

- for all other VRs the Value Length Field is the 16-bit unsigned integer following the two character VR Field (Table 7.1-2). The value of the Value Length Field shall equal the length of the Value Field.

Table 7.1-1. Data Element with Explicit VR of OB, OD, OF, OL, OW, SQ, UC, UR, UT 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", "OD", "OF", "OL", "OW", "SQ", "UC", "UR", "UT" 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

Table 7.1-2. Data Element with Explicit VR other than as shown in Table 7.1-1

Tag		VR	Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 byte character string)	(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