

## DICOM Correction Proposal

STATUS	Final Text
Date of Last Update	2018/11/11
Person Assigned	David Clunie
Submitter Name	Jörg Riesmeier <dicom@jriesmeier.com>
Submission Date	2018-03-05

Correction Number	CP-1799
Log Summary: Fix list of VRs having a VM of 1	
Name of Standard PS3.5 2018d	
Rationale for Correction: The list of Value Representations (VR) having a fixed Value Multiplicity (VM) of one in PS3.5 Section 6.4 is not consistent with the normative specification in Section 6.2: UR is listed but LT, ST and UT are not. The terms "multi-valued" and "multiple valued" are use throughout PS3.5, apparently without any semantic difference. The same term should be used throughout for the concept of data elements having multiple values.	
Correction Wording:	

***Change PS3.5 to use "multi-valued" rather than "multiple valued" everywhere.***

*Change PS3.5 Section 6.4*

### 6.4 Value Multiplicity (VM) and Delimitation

The Value Multiplicity of a Data Element specifies the number of Values that can be encoded in the Value Field of that Data Element. The VM of each Data Element is specified explicitly in [PS3.6](#). If the number of Values that may be encoded in an element is variable, it shall be represented by two numbers separated by a dash; e.g., "1-10" means that there may be 1 to 10 Values in the element.

#### Note

Elements having a multiplicity of "S", which represented "single", in versions of this standard preceding V3.0, will have a multiplicity of "1" in this version of this standard.

When a Data Element has multiple Values, those Values shall be delimited as follows:

- For character strings, the character 5CH (BACKSLASH "\" in the case of the repertoire ISO IR-6) shall be used as a delimiter between Values.

#### Note

BACKSLASH "\" is used as a delimiter between character string Values that are of fixed length as well as variable length.

- Multiple binary Values of fixed length shall be a series of concatenated Values without any delimiter.

Each string Value in a multiple valued character string may be of even or odd length, but the length of the entire Value Field (including "\" delimiters) shall be of even length. If padding is required to make the Value Field of even

length, a single padding character shall be applied to the end of the Value Field (to the last Value), in which case the length of the last Value may exceed the Length of Value by 1.

Note

A padding character may need to be appended to a fixed length character string value in the above case.

Only the last UID Value in a multiple valued Data Element with a VR of UI shall be padded with a single trailing NULL (00H) character when necessary to ensure that the entire Value Field (including "\" delimiters) is of even length.

Data Elements with a VR of LT, OB, OD, OF, OL, OW, SQ, ST, UN, ~~or~~ UR or UT shall always have a Value Multiplicity of one. See Table 6.2-1.