DICOM Change Proposal

STATUS	Assigned
Date of Last Update	2025-01-16
Person Assigned	Clunie
Submitter Name	Jörg Riesmeier <dicom@jriesmeier.com>
Submission Date	2024-09-23

Change Number CP-2464
Log Summary: Clarify empty Value in multi-valued Data Element
Name of Standard PS3.5
Rationale for Change: When reading the first sentence of Note 1 in PS3.5 Section 7.4.1 (see below), one could conclude that only CS, SH and LO are affected by the described situation. However, there does not seem to be a reason why it should not also apply to other string-based VRs such as DS or IS, e.g. if a particular Value is unknown, so it is proposed to add an "e.g." to the VR list in round brackets. It is also proposed to add a sentence on empty Values in multi-valued Data Elements to PS3.5 Section 6.4 (or the like) to make it more explicit what Note 1 in Section 7.4.1 explains (i.e. in a non-normative manner). *Open issues:* • PS3.5 currently uses a mixture of “1” and “one” for specifying the VM (see e.g. below excerpts). It is proposed that one of these two variants be used consistently throughout the Standard text. • If a Data Element Value consisting of “\” delimiters only would be considered an empty Value (see below proposal), shouldn’t a PN Value consisting of “^” and/or “=” only also be treated like an empty Value? Please note that an empty Value is not the same as a Data Element with zero Value Length and no Value (see e.g. PS3.5 Section 7.4.3 in Type 2 requirements).
Change Wording:

Modify PS3.5 Section 6.4 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

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 a Data 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 Data Element.

Note

Elements having a multiplicity of "S", which represented "single", in older versions of this Standard, 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 multi-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 multi-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.

Each string Value in a multi-valued character string may be empty, unless otherwise specified. The presence of one or more delimiter (BACKSLASH) characters alone, without any Values, is treated like an empty Data Element Value, even if the Value Length is greater than zero. See Section 7.4 for details on Data Element Type requirements.

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

Modify PS3.5 Section 7.4.1 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

7.4.1 Type 1 Required Data Elements

IODs and SOP Classes define Type 1 Data Elements that shall be included and are mandatory Data Elements. The Value Field shall contain valid data as defined by the Data Element's VR and VM as specified in PS3.6. The Length of the Value Field shall not be zero. Absence of a valid Value in a Type 1 Data Element is a protocol violation.

Note

1. For Data Elements with a string (**e.g.,** CS, SH, **and** LO) **for which multiple Values are allowed,** rather than binary, text or sequence Value Representation~~, and for which multiple Values are allowed~~, the presence of a single Value is sufficient to satisfy the Type 1 requirement, unless specified otherwise in the Attribute description, and other Values may be empty, unless otherwise specified by the IOD. The presence of one or more delimiter (BACKSLASH) characters alone, without any Values, is not sufficient to satisfy the Type 1 requirement, since even though the Value Length is greater than zero, there is no valid **Data Element** Value present.

2. A Type 1 Sequence Data Element will contain one or more Items, as defined by the IOD (irrespective of the VM of the Sequence, which is always ~~one~~**1** (Section 7.5)). Whether or not those Items may be empty (contain no Data Elements) depends on the IOD definition of the Data Set for each Item.

Commented [DC1]: Clarify exactly what VRs this applies to.