Status	Letter Ballot
Date of Last Update	2026/01/28
Person Assigned	David Clunie mailto:dclunie@dclunie.com
Submitter Name	David Clunie mailto:dclunie@dclunie.com
Submission Date	2023/08/01

Correction Number CP-2332
Log Summary: Bulk data value is not intended for Sequences
Name of Standard PS3.18, PS3.19
Rationale for Correction: The standard does not specify explicitly that the XML or JSON bulk data value returned as URI is limited to encoding data elements with an actual Value Field, rather than also applying to "a Value consisting of a Sequence of zero or more Items". Add such a restriction. There is already a restriction for the XML in-line data to be of OB, OD, OF, OL, OV, OW, or UN, though this is not present in the JSON description. For PS3.18, reference the PS3.19 description rather than just repeating parts of it. One could propagate the same restriction for the in-line data to the URI data, which would then exclude SQ, but that would also prohibit the use of VRs like US, which are used for lookup tables such as that already exemplified. So, just forbid SQ VR. This restriction will not apply to PixelData of undefined length, which uses a Sequence-like structure to encode its value, but is not a Sequence per se. Alternatively, one might argue that since the SQ mechanism for the URI has not actually been forbidden, it is possible, and that it might be useful (e.g, to elide but provide a means of retrieval for a large Per-Frame Functional Groups Sequence), but that would beg the question of the Transfer Syntax of what was encoded, which isn't specified. Whether the Transfer Syntax is Implicit or Explicit VR affects the encoded bytes. Since this seems like an unlikely interpretation, its addition would likely break existing user agents, which would not be expecting it. *[Ed.Note: Another question is which byte order applies, since the Transfer Syntax is not specified (re. "byte ordering and swapping").*
Correction Wording:

*Amend DICOM PS3.19 as follows (changes to existing text are bold and **underlined** for additions and **~~struckthrough~~** for removals):*

A.1.5 Description

Table A.1.5-2. DICOM Data Set Macro

Name	Optionality	Cardinality	Description
DicomAttribute	O	0-n	An Infoset element corresponding to each DICOM Attribute.
...			
>BulkData	C	1	A reference to a blob of data that the recipient may retrieve through use of the GetData() method, a PS3.18 Studies Service Retrieve (WADO-RS) transaction or a PS3.18 Studies Service Store (STOW-RS) transaction. Required if the DICOM Data Element represented is not zero length and an XML Infoset Value, Item, InlineBinary or PersonName element is not present. The provider of the data may use a BulkData reference at its discretion to avoid encoding a large DICOM Value Field as text by value in the Infoset. For example, pixel data or look up tables. There is a single BulkData Infoset element representing the entire Value Field, and not one per Value in the case where the Value Multiplicity is greater than one. **Note** E.g., a LUT with 4096 16 bit entries that may be encoded in DICOM with a Value Representation of OW, with a VL of 8192 and a VM of 1, or a US VR with a VL of 8192 and a VM of 4096 would both be represented as a single BulkData element. All rules (e.g., byte ordering and swapping) in PS3.5 apply. **Note** Implementers should pay particular attention to the PS3.5 rules regarding the value representations of OD, OF, OL, OV and OW. If the BulkData has a string or text Value Representation, the value(s) of the DICOM Specific Character Set Data Element, if present, might be necessary to determine its encoding. **Shall not be used for a Data Element with SQ VR.**
>>uuid	C	A	An identifier of this bulk data reference formatted as a UUID using the hexadecimal representation defined in ???. Required if BulkData URI is not present. Shall not be present otherwise.
>>uri	C	A	The HTTP(S) URI for this bulk data reference. Required if the NativeDicomModel was: • returned in response to a PS3.18 Studies Service Retrieve (WADO-RS) Retrieve Metadata request Shall not be present otherwise.

Name	Optionality	Cardinality	Description
>InlineBinary	C	1	The Value Field of the enclosing Attribute encoded as base64. Required if the DICOM Data Element represented is: • not zero length • the VR if the enclosing Attribute is OB, OD, OF, OL, OV, OW, or UN • an XML Infoset Value or BulkData XML element is not present Shall not be present otherwise. There is a single InlineBinary Infoset element representing the entire Value Field, and not one per Value in the case where the Value Multiplicity is greater than one. **Note** E.g., a LUT with 4096 16 bit entries that may be encoded in DICOM with a Value Representation of OW with a VL of 8192 and a VM of 1 would be represented as a single InlineBinary element. All rules (e.g., byte ordering and swapping) in PS3.5 apply. **Note** Implementers should pay particular attention to the PS3.5 rules regarding the value representations of OD, OF, OL, OV and OW.

*Amend DICOM PS3.18 as follows (changes to existing text are bold and **underlined** for additions and ~~**struckthrough**~~ for removals):*

F.2 DICOM JSON Model

F.2.6 BulkDataURI

If an attribute contains a "BulkDataURI" , this contains the URI of a bulk data element as defined in Table A.1.5-2 in PS3.19.

F.2.7 InlineBinary

If an attribute contains an "InlineBinary", this contains the base64 encoding of the enclosing attribute's Value Field **as defined in Table A.1.5-2 in PS3.19**.

~~**There is a single InlineBinary value representing the entire Value Field, and not one per Value in the case where the Value Multiplicity is greater than one. E.g., a LUT with 4096 16 bit entries that may be encoded in DICOM with a Value Representation of OW, with a VL of 8192 and a VM of 1, or a US VR with a VL of 8192 and a VM of 4096 would both be represented as a single InlineBinary string.**~~

~~**All rules (e.g., byte ordering and swapping) in DICOM PS3.5 apply.**~~

Note

~~**Implementers should in particular pay attention to the PS3.5 rules regarding the value representations of OD, OF, OL and OW.**~~

For reference, from DICOM PS3.5 unchanged:

7.1.1 Data Element Fields

A Data Element is made up of fields. Three fields are common to all three Data Element structures; these are the Data Element Tag, Value Length, and Value Field. A fourth field, Value Representation, is only present in the two Explicit VR Data Element structures. The Data Element structures are defined in ??? and ???. The definitions of the fields are:

Data Element Tag An ordered pair of 16-bit unsigned integers representing the Group Number followed by Element Number.

Value Representation Two single byte characters containing the VR of the Data Element. The VR for a given Data Element Tag shall be as defined by the Data Dictionary as specified in PS3.6. The two byte VR shall be encoded using only upper case letters from the DICOM default character set.

Value Length Either:

- a 16 or 32-bit (dependent on VR and whether VR is explicit or implicit) unsigned integer containing the Explicit Length of the Value Field as the number of bytes (even) that make up the Value. It does not include the length of the Data Element Tag, Value Representation, and Value Length Fields.
- a 32-bit Value Length Field set to Undefined Length (FFFFFFFFH). Undefined Lengths may be used for Data Elements having the Value Representation (VR) Sequence of Items (SQ) and Unknown (UN). For Data Elements with Value Representation OW or OB Undefined Length may be used depending on the negotiated Transfer Syntax (see ??? and ???).

Note

1. The decoder of a Data Set should support both Explicit and Undefined Lengths for VRs of SQ and UN and, when applicable, for VRs of OW and OB.
2. The 32-bit Value Length Field limits the maximum size of large data Value Fields such as Pixel Data sent in a Native Format (encoded in Transfer Syntaxes that use only the unencapsulated form).

Value Field An even number of bytes containing the Value(s) of the Data Element.

The data type of Value(s) stored in this field is specified by the Data Element's VR. The VR for a given Data Element Tag can be determined using the Data Dictionary in PS3.6, or using the VR Field if it is contained explicitly within the Data Element. The VR of Standard Data Elements shall agree with those specified in the Data Dictionary.

The Value Multiplicity specifies how many Values with this VR can be placed in the Value Field. If the VM is greater than one, multiple Values shall be delimited within the Value Field as defined previously in ???. The VMs of Standard Data Elements are specified in the Data Dictionary in PS3.6.

Value Fields with Undefined Length are delimited through the use of Sequence Delimitation Items and Item Delimitation Data Elements, which are described further in Section 7.5.

7.5 Nesting of Data Sets

The VR identified "SQ" shall be used for Data Elements with a Value consisting of a Sequence of zero or more Items, where each Item contains a set of Data Elements. SQ provides a flexible encoding scheme that may be used for simple structures of repeating sets of Data Elements, or the encoding of more complex Information Object Definitions often called folders. SQ Data Elements can also be used recursively to contain multi-level nested structures.

Items present in an SQ Data Element shall be an ordered set where each Item may be referenced by its ordinal position. Each Item shall be implicitly assigned an ordinal position starting with the Value 1 for the first Item in the Sequence, and incremented by 1 with each subsequent Item. The last Item in the Sequence shall have an ordinal position equal to the number of Items in the Sequence.

Note

1. This clause implies that item ordering is preserved during transfer and storage.

2. An IOD or Module Definition may choose to not use this ordering property of a Data Element with VR of SQ. This is simply done by not specifying any specific semantics to the ordering of Items, or by not specifying usage of the referencing of Items by ordering position.

The definition of the Data Elements encapsulated in each Item is provided by the specification of the Data Element (or associated Attribute) of Value Representation SQ. Items in a sequence of Items may or may not contain the same set of Data Elements. Data Elements with a VR of SQ may contain multiple Items but shall always have a Value Multiplicity of one (i.e., a single Sequence).

There are three special SQ related Data Elements that are not ruled by the VR encoding rules conveyed by the Transfer Syntax. They shall be encoded as Implicit VR. These special Data Elements are Item (FFFE,E000), Item Delimitation Item (FFFE,E00D), and Sequence Delimitation Item (FFFE,E0DD). However, the Data Set within the Value Field of the Data Element Item (FFFE,E000) shall be encoded according to the rules conveyed by the Transfer Syntax.

7.5.1 Item Encoding Rules

Each Item of a Data Element of Value Representation SQ shall be encoded as a DICOM Standard Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Value (Item) Length field encoded in one of the following two ways:

a. Explicit Length: The number of bytes (even) contained in the Sequence Item Value (following but not including the Value (Item) Length Field) is encoded as a 32-bit unsigned integer value (see ???). This length shall include the total length of all Data Elements conveyed by this Item. This Value (Item) Length Field shall be equal to 00000000H if the Item contains no Data Set.

b. Undefined Length: The Value (Item) Length Field shall contain the value FFFFFFFFH to indicate an Undefined Length. It shall be used in conjunction with an Item Delimitation Data Element. This Item Delimitation Data Element has a Data Element Tag of (FFFE,E00D) and shall follow the Data Elements encapsulated in the Item. No Value shall be present in the Item Delimitation Data Element and its Value (Item) Length shall be 00000000H. An Item containing no Data Set is encoded by an Item Delimitation Data Element only.

The encoder of a Data Set may choose either one of the two ways of encoding. Both ways of encoding shall be supported by decoders of Data Sets. Data Element Tags (FFFF,eeee) are reserved by this Standard and shall not be used.

Each Item Value shall contain a DICOM Data Set composed of Data Elements. Within the context of each Item, these Data Elements shall be ordered by increasing Data Element Tag value and appear only once (as Data Set is defined in ???). There is no relationship between the ordering of the Data Elements contained within an Item and the ordering of the Data Element Tag of SQ Value Representation that contains that Item. One or more Data Elements in an Item may be of Value Representation SQ, thus allowing for recursion.

Data Elements with a group of 0000, 0002 and 0006 shall not be present within Sequence Items.

Note

The use of Transfer Syntax UID (0002,0010) in particular is forbidden, since were it to differ from the Transfer Syntax of the enclosing Data Set then a change in encoding would be implied, which is not allowed.

??? specifies rules for incorporating Private Data Elements into Sequence Items.

7.5.2 Delimitation of The Sequence of Items

Delimitation of the last Item of a Sequence of Items, encapsulated in a Data Element of Value Representation SQ, shall be in one of the two following ways:

a. Explicit Length: The number of bytes (even) contained in the Data Element Value (following but not including the Value (Sequence) Length Field) is encoded as a 32-bit unsigned integer value (see ???). This length shall include the total length resulting from the sequence of zero or more items conveyed by this Data Element. This Data Element Length shall be equal to 00000000H if the sequence of Items contains zero Items.

b. Undefined Length: The Value (Sequence) Length Field shall contain a Value FFFFFFFFH to indicate a Sequence of Undefined Length. It shall be used in conjunction with a Sequence Delimitation Item. A Sequence Delimitation Item shall be included after the last Item in the sequence. Its Item Tag shall be (FFFE,E0DD) with a Value (Item) Length Field of 00000000H. No Value shall be present. A Sequence containing zero Items is encoded by a Sequence Delimitation Item only.

The encoder of a Sequence of Items may choose either one of the two ways of encoding. Both ways of encoding shall be supported by decoders of the Sequence of Items.

Note

The Sequence Delimitation Item Tag (FFFE,E0DD) is different from the Item Delimitation Tag (FFFE,E00D) introduced above in that it indicates the end of a Sequence of Items whose Length was left undefined. If an Undefined Length Item is the last Item of a Sequence of Items of Undefined Length, then an Item Delimitation Tag will be followed by a Sequence Delimitation Tag.

For an example of an SQ Data Element of Explicit Length encapsulating Items of Explicit Length see Table 7.5-1.

For an example of an SQ Data Element of Undefined Length encapsulating Items of Explicit Length see Table 7.5-2.

For an example of an SQ Data Element of Undefined Length encapsulating Items of both Explicit and Undefined Length see Table 7.5-3.

Table 7.5-1. Example of a Data Element with Implicit VR Defined as a Sequence of Items (VR = SQ) with Three Items of Explicit Length

Data Element Tag	**Data Element Length**	**Data Element Value**								
(gggg, eeee) with VR of SQ	0000 0F00H	**First Item**			**Second Item**			**Third Item**		
		Item Tag (FFFE, E000)	Item Length 0000 04F8H	Item Value Data Set	Item Tag (FFFE, E000)	Item Length 0000 04F8H	Item Value Data Set	Item Tag (FFFE, E000)	Item Length 0000 04F8H	Item Value Data Set
4 bytes	4 bytes	4 bytes	4 bytes	04F8H bytes	4 bytes	4 bytes	04F8H bytes	4 bytes	4 bytes	04F8H bytes

Table 7.5-2. Example of a Data Element with Explicit VR Defined as a Sequence of Items (VR = SQ) of Undefined Length, Containing Two Items of Explicit Length

Data Element Tag	**Value Representation**		**Data Element Length**	**Data Element Value**							
(gggg, eeee) with VR of SQ	SQ	0000H Reserved	FFFF FFFFH Undefined Length	**First Item**			**Second Item**			**Sequence Delimitation Item**	
				Item Tag (FFFE, E000)	Item Length 98A5 2C68H	Item Value Data Set	Item Tag (FFFE, E000)	Item Length B321 762CH	Item Value Data Set	Seq. Delim. Tag (FFFE, E0DD)	Item Length 0000 0000H
4 bytes	2 bytes	2 bytes	4 bytes	4 bytes	4 bytes	98A5 2C68H bytes	4 bytes	4 bytes	B321 762CH bytes	4 bytes	4 bytes

Note

The Data Set within the Item Values in Table 7.5-2 have VRs Explicitly defined.

Table 7.5-3. Example of a Data Element with Implicit VR Defined as a Sequence of Items (VR = SQ) of Undefined Length, Containing Two Items Where One Item is of Explicit Length and the Other Item is of Undefined Length

Data Element Tag	**Data Element Length**	***Data Element Value***									
(gggg, eeee) with VR of SQ	FFFF FFFFH Undefined Length	**First Item**			**Second Item**					**Sequence Delimitation Item**	
		Item Tag (FFFE, E000)	Item Length 0000 17B6H	Item Value Data Set	Item Tag (FFFE, E000)	Item Length FFFF FFFFH Undefined Length	Item Value Data Set	Item Delim. Tag (FFFE, E00D)	Length 0000 0000H	Seq. Delim. Tag (FFFE, E0DD)	Item Length 0000 0000H
4 bytes	4 bytes	4 bytes	4 bytes	17B6H bytes	4 bytes	4 bytes	Undefined Length	4 bytes	4 bytes	4 bytes	4 bytes