

1	Status	Assigned
2	Date of Last Update	2019/05/16
3	Person Assigned	David Clunie
4		mailto:dclunie@dclunie.com
5	Submitter Name	David Clunie
6		mailto:dclunie@dclunie.com
7	Submission Date	2019/02/22

8	Correction Number CP-1900
9	Log Summary: WADO-RS Transfer Syntax to use when cannot be decompressed
10	Name of Standard
11	PS3.5, PS3.18
12	Rationale for Correction:
13	It has long been recognized that images that are stored in lossy compressed form need not be decompressed for retrieval, and more
14	recently it has been understood that extremely large compressed images (lossy or lossless) may not be representable in the default
15	Implicit VR uncompressed Transfer Syntax, or for that matter in the Explicit VR uncompress Transfer Syntax. Futher, there are certain
16	attributes that may be stored in Implicit VR that are not representable in Explicit VR due to shorter value length representation.
17	The default of Explicit VR specified for WADO-RS is therefore inappropriate for such cases, and should be modified to account for
18	these situations. The description of the WADO URI query parameter already partially addresses some of these concerns.
19	These corrections are particular important for enterprise imaging applications, where the image pixel data is very often significantly
20	lossy compressed and return in its uncompressed form would be impractical and ludicrous.
21	Note that the user agent does not know in what form the origin server has stored or has access to the image, so it cannot be assumed
22	that it will be able to make the right choice.
23	<i>[Ed.Note: Similar concerns (what to include if not specified) will likely also apply to Sup 211 packaging in application/zip.]</i>
24	<i>[Ed.Note: Have any changes required been lost in converting from pre-Sup 183 redocumentation to current PS3.18? Is any further</i>
25	<i>simplification/refactoring required, since the same phrase is repeated in multiple locations? The definition and use of "default" in</i>
26	<i>particular remains somewhat confusing.]</i>
27	<i>[Ed.Note: Do the circumstances under which a 406 error will be returned need to be defined, e.g., request EVRLE but not possible</i>
28	<i>to convert it, etc. ?]</i>
29	<i>[Ed.Note: What about lossless images that are also huge and unrealistic to decompress but could be, e.g., WSI J2K reversible tiles?]</i>
30	Correction Wording:

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

8.7.3 DICOM Media Types and Media Types For Bulkdata

This section defines the media types used to represent DICOM Instances, Metadata and Bulkdata. It describes:

- The media type and Transfer Syntax parameters for DICOM Instance
- The media types that can be used for Metadata
- The media types and Transfer Syntaxes parameters for Bulkdata
- The syntax of DICOM Media Types including their Transfer Syntax and character set parameters
- The Query Parameter for Transfer Syntax
- The meaning of Acceptable Transfer Syntaxes and Selected Transfer Syntax

The media types defined in this section are distinct from those into which DICOM Instances may be rendered (which are defined in ???); some of the same media types are used for both rendered content and Bulkdata.

Depending on the service, the media types may be single part or multipart, and may have required or optional Transfer Syntax and/or character set parameters.

The Implicit VR Little Endian (1.2.840.10008.1.2), and Explicit VR Big Endian (1.2.840.10008.1.2.2 - Retired) Transfer Syntaxes shall not be used with Web Services.

If a Transfer Syntax parameter for a DICOM Media Type is not specified in a request or response, the Transfer Syntax in the response shall be the Transfer Syntax specified as the default for the Resource Category and media type combination in Table 8.7.3-2, Table 8.7.3-4 or Table 8.7.3-5, **unless the origin server has only access to the pixel data in lossy compressed form or the pixel data in a lossless compressed form that is of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.**

Table 8.7.3-1 specifies the definition of media type requirement terms used in the tables in this section.

Table 8.7.3-1. Definition of Media Type Requirement

Requirement	Optionality	Definition
Default	D	The origin server shall return this media type when none of the Acceptable Media Types (see ???) are supported. The origin server shall support this media type.
Required	R	The origin server shall support this media type.
Optional	O	The origin server may support this media types.

Table 8.7.3-2, Table 8.7.3-3, Table 8.7.3-4, and Table 8.7.3-5 specify the media types used to encode different representations of DICOM Instances. These media types apply to all Resource Categories and have default encodings for images and video data elements contained in the Instances.

8.7.3.1 The application/dicom Media Type

The application/dicom media type specifies a representation of Instances encoded in the DICOM File Format specified in ????

Table 8.7.3-2 specifies the default and optional Transfer Syntax UID combinations for each DICOM Resource Category (see ???). The default media type for the Resource Category shall be returned when the origin server supports none of the Acceptable Media Types, **unless the origin server has only access to the pixel data in lossy compressed form or the pixel data in a lossless compressed form that is of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.**

Table 8.7.3-2. Transfer Syntax UIDs for application/dicom Media Types

Category	Transfer Syntax UID	Transfer Syntax Name	Optionality
Single Frame Image	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
	1.2.840.10008.1.2.4.70	JPEG Lossless, Non-Hierarchical, First-Order Prediction(Process 14 [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression	O
	1.2.840.10008.1.2.4.50	JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression	O
	1.2.840.10008.1.2.4.51	JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)	O
	1.2.840.10008.1.2.4.57	JPEG Lossless, Non-Hierarchical (Process 14)	O
	1.2.840.10008.1.2.5	RLE Lossless	O
	1.2.840.10008.1.2.4.80	JPEG-LS Lossless Image Compression	O
	1.2.840.10008.1.2.4.81	JPEG-LS Lossy (Near-Lossless) Image Compression	O
	1.2.840.10008.1.2.4.90	JPEG 2000 Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.91	JPEG 2000 Image Compression	O
	1.2.840.10008.1.2.4.92	JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)	O
1.2.840.10008.1.2.4.93	JPEG 2000 Part 2 Multi-component Image Compression	O	
Multi-frame Image	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
	1.2.840.10008.1.2.4.90	JPEG 2000 Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.91	JPEG 2000 Image Compression	O
	1.2.840.10008.1.2.4.92	JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)	O
	1.2.840.10008.1.2.4.93	JPEG 2000 Part 2 Multi-component Image Compression	O
Video	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
	1.2.840.10008.1.2.4.100	MPEG2 Main Profile @ Main Level	O
	1.2.840.10008.1.2.4.101	MPEG2 Main Profile @ High Level	O
	1.2.840.10008.1.2.4.102	MPEG-4 AVC/H.264 High Profile / Level 4.1	O
	1.2.840.10008.1.2.4.103	MPEG-4 AVC/H.264 BD-compatible High Profile / Level 4.1	O
	1.2.840.10008.1.2.4.104	MPEG-4 AVC/H.264 High Profile / Level 4.2 For 2D Video	O
	1.2.840.10008.1.2.4.105	MPEG-4 AVC/H.264 High Profile / Level 4.2 For 3D Video	O
	1.2.840.10008.1.2.4.106	MPEG-4 AVC/H.264 Stereo High Profile / Level 4.2	O
Text	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
Other	1.2.840.10008.1.2.1	Explicit VR Little Endian	D

8.7.3.2 DICOM Metadata Media Types

Table 8.7.3-3 specifies the media types that may be used to encode representations of Metadata for the URI and RESTful services. Only the RESTful Services support Metadata representations.

Table 8.7.3-3. Media Types for Metadata

Media Type	Descriptions	URI	RESTful
application/dicom+xml	Encodes Instances as XML Infosets defined in the Native DICOM Model defined in ?????.	not applicable	required

Media Type	Descriptions	URI	RESTful
application/dicom+json	Encodes Instances in the JSON format defined in ???.	not applicable	required

8.7.3.3 DICOM Bulkdata Media Types

Bulkdata representations are only supported by RESTful services. There are two categories of Bulkdata: uncompressed and compressed.

The default media type for the Resource Category shall be returned when the origin server supports none of the Acceptable Media Types, unless the origin server has only access to the pixel data in lossy compressed form or the pixel data in a lossless compressed form that is of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.

The origin server may support additional Transfer Syntaxes.

If no media type Transfer Syntax parameter is specified, then the Explicit VR Little Endian Transfer Syntax "1.2.840.10008.1.2.1" shall be used, unless the origin server has only access to the pixel data in lossy compressed form or the pixel data in a lossless compressed form that is of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.

Note

The tables in this section have no entries for the URI service, since they do not support separate retrieval of Bulkdata.

8.7.3.3.1 Uncompressed Bulkdata

Table 8.7.3-4 specifies the default media type and Transfer Syntax UIDs, by Resource Category (see ???) that can be used with uncompressed Bulkdata for the RESTful services. Uncompressed Bulkdata is encoded as a stream of uncompressed bytes (octets) in Little Endian byte order.

Note

This is the same encoding defined in ??? for the returned value of the getData() call for uncompressed Bulkdata.

Table 8.7.3-4. Transfer Syntax UIDs for Uncompressed Data in Bulkdata

Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	RESTful
Single Frame Image	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
Multi-Frame Image	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
Video	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
Text	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
Other	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D

Note

Even though the Transfer Syntax is Explicit VR Little Endian, the Value Representation is not actually encoded at the beginning of the octet-stream. The Value Representation is contained in the Metadata that references the Bulkdata.

8.7.3.3.2 Compressed Bulkdata

Compressed Bulkdata contains only the compressed octet stream without the fragment delimiters.

Table 8.7.3-5 specifies the default and optional media types and Transfer Syntax UID combinations for each Resource Category (see ???) of compressed Bulkdata for the RESTful services.

Note

Some of the Transfer Syntax Names include text about Default Transfer Syntax, however this applies to its role in DIMSE transactions, rather than the default for RESTful services (which is specified in the RESTful column of the table).

1 These media types can be used to retrieve Bulkdata, such as images or video, encoded in a specific Transfer Syntax.

2 **Table 8.7.3-5. Media Types and Transfer Syntax UIDs for Compressed Data in Bulkdata**

3 Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	Optionality
4 Single 5 Frame 6 Image	image/jpeg	1.2.840.10008.1.2.4.70	JPEG Lossless, Non-Hierarchical, First-Order Prediction(Process 14 [Selection Value 1]) :Default Transfer Syntax for Lossless JPEG Image Compression	D
		1.2.840.10008.1.2.4.50	JPEG Baseline (Process 1) :Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression	O
		1.2.840.10008.1.2.4.51	JPEG Extended (Process 2 & 4) :Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)	O
		1.2.840.10008.1.2.4.57	JPEG Lossless, Non-Hierarchical (Process 14)	O
	image/x-dicom-rle	1.2.840.10008.1.2.5	RLE Lossless	D
	image/x-jls	1.2.840.10008.1.2.4.80	JPEG-LS Lossless Image Compression	D
		1.2.840.10008.1.2.4.81	JPEG-LS Lossy (Near-Lossless) Image Compression	O
	image/jp2	1.2.840.10008.1.2.4.90	JPEG 2000 Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.91	JPEG 2000 Image Compression	O
	image/jpx	1.2.840.10008.1.2.4.92	JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.93	JPEG 2000 Part 2 Multi-component Image Compression	O
	23 Multi-frame 24 Image	image/jpeg	1.2.840.10008.1.2.4.70	JPEG Lossless, Non-Hierarchical, First-Order Prediction(Process 14 [Selection Value 1]) :Default Transfer Syntax for Lossless JPEG Image Compression
1.2.840.10008.1.2.4.50			JPEG Baseline (Process 1) :Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression	O
1.2.840.10008.1.2.4.51			JPEG Extended (Process 2 & 4) :Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)	O
1.2.840.10008.1.2.4.57			JPEG Lossless, Non-Hierarchical (Process 14)	O
image/x-dicom-rle		1.2.840.10008.1.2.5	RLE Lossless	D
image/x-jls		1.2.840.10008.1.2.4.80	JPEG-LS Lossless Image Compression	D
		1.2.840.10008.1.2.4.81	JPEG-LS Lossy (Near-Lossless) Image Compression	O
image/jp2		1.2.840.10008.1.2.4.90	JPEG 2000 Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.91	JPEG 2000 Image Compression	O
image/jpx		1.2.840.10008.1.2.4.92	JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)	D
		1.2.840.10008.1.2.4.93	JPEG 2000 Part 2 Multi-component Image Compression	O
42 Video		video/mpeg2	1.2.840.10008.1.2.4.100	MPEG2 Main Profile @ Main Level
	1.2.840.10008.1.2.4.101		MPEG2 Main Profile @ High Level	D
	video/mp4	1.2.840.10008.1.2.4.102	MPEG-4 AVC/H.264 High Profile / Level 4.1	D
		1.2.840.10008.1.2.4.103	MPEG-4 AVC/H.264 BD-compatible High Profile / Level 4.1	O

Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	Optionality
		1.2.840.10008.1.2.4.104	MPEG-4 AVC/H.264 High Profile / Level 4.2 For 2D Video	O
		1.2.840.10008.1.2.4.105	MPEG-4 AVC/H.264 High Profile / Level 4.2 For 3D Video	O
		1.2.840.10008.1.2.4.106	MPEG-4 AVC/H.264 Stereo High Profile / Level 4.2	O
Text		N/A (no defined compression transfer syntaxes for Text)		
Other		N/A (no defined compression transfer syntaxes for Other)		

The origin server may support additional Transfer Syntaxes.

Note

- For the media type image/jpeg Transfer Syntaxes, the image may or may not include the JFIF marker segment. The image may or may not include APP2 marker segments with an identifier of "ICC_PROFILE". There is no requirement for the origin server to add a JFIF marker segment nor to copy the value of the ICC Profile (0028,2000) Attribute, if present, into APP2 marker segments in the compressed data stream. See ????
 - For the media type image/dicom+jp2 and image/dicom+jpx Transfer Syntaxes, the image does not include the jp2 marker segment. See ???? and ????
 - The resource on the origin server may have been encoded in the Deflated Explicit VR Little Endian (1.2.840.10008.1.2.1.99) Transfer Syntax. If so, the origin server may inflate it, and then convert it into an Acceptable Transfer Syntax. Alternatively, if the user agent allowed a Content-Encoding header field of 'deflate', then the deflated bytes may be transferred unaltered, but the Transfer Syntax parameter in the response should be the Explicit VR Little Endian Transfer Syntax.
 - Compressed multi-frame image Bulkdata is encoded as one frame per part. E.g., each frame of a JPEG 2000 multi-frame image will be encoded as a separate part with an image/jp2 media type, rather than as a single part with a video/mj2 (???) or uncompressed application/octet-stream media type.
 - Video Bulkdata is encoded as a single part containing all frames. E.g., all frames of an MPEG-4 video will be encoded as a single part with a video/mp4 (???) media type.
 - Many of the media types used for compressed Pixel Data transferred as Bulkdata values are also used for consumer format media types. A web browser may not be able to display the encoded data directly, even though some of the same media types are also used for encoding rendered Pixel Data. See ???.
- For example, the media type for Bulkdata values of lossless 16-bit JPEG ??? encoded Pixel Data is "image/jpeg", the same media type as might be used for 8-bit JPEG ??? encoded Pixel Data, whether extracted as Bulkdata, or rendered. The Transfer Syntax parameter of the Content-Type header field is useful to signal the difference.
- Each part of a multipart response is distinguished by the Content-Type and Content-Location header fields of the part.

8.7.3.4 Transfer Syntax

The Default Transfer Syntax for DICOM objects contained in a payload shall be Explicit VR Little Endian Uncompressed "1.2.840.10008.1.2.1". If the Transfer Syntax is not specified in a message, then the Default Transfer Syntax shall be used, **unless the origin server has only access to the pixel data in lossy compressed form or the pixel data in a lossless compressed form that is of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.**

Note

- This is different from the Default Transfer Syntax defined in ????, which is Implicit VR Little Endian.
- In other words, every origin server is required to be able to convert any Data Set it is going to return into the Explicit VR Little Endian Transfer Syntax, regardless of the form in which it originally received or stored the Data Set, except in the cases of when the decompressed Pixel Data is too large to encode in the Explicit VR Little Endian Transfer Syntax or is received in a lossy compressed form. In the case of lossy compressed Pixel**

Data, the origin server is permitted to return the lossy compressed Transfer Syntax appropriate to the lossy form that was received. In the case of lossless compressed Pixel Data that is too large to encode in the Explicit VR Little Endian Transfer Syntax, the origin server is permitted to return any appropriate lossless compression Transfer Syntax, not necessarily that in which the image was received, as an alternative to the Explicit VR Little Endian Transfer Syntax.

3. **If transcoding to the Explicit VR Little Endian Transfer Syntax, a VR of UN may be needed for the encoding of Data Elements with explicit VR whose value length exceeds 65534 ($2^{16}-2$) (FFFEH, the largest even length unsigned 16 bit number) but which are defined to have a 16 bit explicit VR length field. See ????.**

Implicit VR Little Endian, or Explicit VR Big Endian shall not be used.

The response payload encoding requirements are defined in ???.

Note

The transfer syntax can be one of the JPIP Transfer Syntaxes, in which case the returned objects will contain the URL of the JPIP provider for retrieving the pixel data.

The origin server may support additional Transfer Syntaxes.

8.7.3.5 DICOM Media Type Syntax

The syntax of DICOM Media Types is:

`dicom-media-type = (dicom-segment / dicom-multi-part) [dicom-parameters]`

Where

`dicom-segment = dicom-rt-name`

`dicom-multi-part` ; see Section 8.7.3.5.1

`dicom-parameters = transfer-syntax-ntp` ; see Section 8.7.3.5.2

`/ charset-ntp`; see Section 8.7.3.5.3

`dicom-rt-name = dicom / dicom+xml / dicom+json` ; DICOM Media Type name

`dicom = "application/dicom"`

`dicom+xml = "application/dicom+xml"`

`dicom+json = "application/dicom+json"`

`octet-stream = "application/octet-stream"`

All DICOM Media Types may have a Transfer Syntax parameter, but its usage may be constrained by the service for which they are used.

Note

The application/dicom+xml and application/dicom+json Media Types may have a Transfer Syntax parameter in order to specify the encoding of base64 data.

All DICOM Media Types may have a character set parameter, but its usage may be constrained by the service for which they are used.

8.7.3.5.1 DICOM Multipart Media Types

The syntax of multipart media types is:

`dicom-multi-part = "multipart/related"`

1 O/S ";" O/S "type" "=" dcm mp-rt-name

2 O/S ";" O/S "boundary=" boundary

3 [dcm parameters]

4 [related-parameters]

5 Where

6 dcm mp-rt-name = di com / di com xnh / di com j son / octet-stream

7 See ??? for the definition of boundary and related-parameters.

8 Each multipart media type shall include a "type" parameter that defines the media type of the parts and shall also include a "boundary"
9 parameter that specifies the boundary string that is used to separate the parts.

10 8.7.3.5.2 Transfer Syntax Parameter

11 For a given DICOM Media Type, a single Transfer Syntax parameter value may be specified, but its usage may be constrained by
12 the service for which they are used.

13 RESTful origin servers shall support the Transfer Syntax parameter.

14 Transfer syntax media type parameters are forbidden in URI Service requests and responses.

15 The syntax is:

16 transfer-syntax-ntp = O/S ";" O/S %s"transfer-syntax=" ts-value

17 ts-value = transfer-syntax-uid / "*"

18 transfer-syntax-uid ; a UID from???? with a UID Type of Transfer Syntax

19 The value of the Transfer Syntax parameter may be either a Transfer Syntax UID or the token "*".

20 For example, to specify that 1.2.840.10008.1.2.4.50 is the acceptable Transfer Syntaxes, an Accept header field could be:

21 Accept: appli cati on/di com transfer-syntax=1.2.840.10008.1.2.4.50

22 A DICOM Media Type may only have one Transfer Syntax parameter and it shall have only one value.

23 Note

24 Per ??? Media Type Specifications and Registration Procedures, it is an error for a specific parameter to be specified more
25 than once. If a choice of Transfer Syntaxes is acceptable, more than one media type may be provided in the Accept header
26 with different q parameter values to indicate preference. E.g., to specify that 1.2.840.10008.1.2.4.50 and to specify that
27 1.2.840.10008.1.2.4.57 are acceptable but 1.2.840.10008.1.2.4.50 is preferred, an Accept header field could be:

28 Accept: mul ti part/rel ated; type="appli cati on/di com"; transfer-syntax=1.2.840.10008.1.2.4.50; boundary=**, mul ti part/rel ated

29 The wildcard value "*" indicates that the user agent will accept any Transfer Syntax. This allows, for example, the origin server to respond
30 without needing to transcode an existing representation to a new Transfer Syntax, or to respond with the Explicit VR Little Endian
31 Transfer Syntax regardless of the Transfer Syntax stored, **unless the origin server has only access to the pixel data in lossy
32 compressed form or the pixel data in a lossless compressed form that is of such length that it cannot be encoded in the
33 Explicit VR Little Endian Transfer Syntax.**

34 If an Origin server supports the Transfer Syntax parameter, it shall support the wildcard value.

35 Origin servers that support the Transfer Syntax parameter shall specify in their Conformance Statement those values of Transfer
36 Syntax parameter that are supported in the response.

User agents that support the Transfer Syntax parameter shall specify in their Conformance Statement those Transfer Syntax parameter values that may be supplied in the request.

8.7.3.5.3 Character Set Parameter

All DICOM Media Types may have a single Character Set parameter, which shall have only a single value, that specifies the Acceptable Character Set for the response.

The syntax is:

```
charset-rtpp = 0% "; " 0% %"charset" "=" charset
```

All DICOM Media Types have a Default Character Set of UTF-8. See ??? for character set details.

8.7.3.6 Transfer Syntax Query Parameter

The Transfer Syntax Query Parameter specifies a comma-separated list of one or more Transfer Syntax UIDs, as defined in ????

The syntax is:

```
transfer-syntax-qp = %"transferSyntax" "=" (1#transfer-syntax-uid / "**")
```

This Query Parameter is only used by the URI Service.

RESTful services specify the Transfer Syntax in the "accept" Query Parameter (see ???) and do not use Transfer Syntax Query Parameter.

8.7.3.7 Acceptable Transfer Syntaxes

Each DICOM Media Type in the Acceptable Media Types has an Acceptable Transfer Syntax, which is explicitly specified or has a default value.

Depending on the service, the Acceptable Transfer Syntax for a DICOM Media Type can be specified in the:

1. Transfer Syntax media type parameter contained in the Accept Query Parameter (see ???)
2. Transfer Syntax media type parameter contained in the Accept header field
3. Transfer Syntax Query Parameter (see Section 8.7.3.5)

Amend DICOM PS3.5 as follows (changes to existing text are bold and underlined for additions and ~~struckthrough~~ for removals):

6.2.2 Unknown (UN) Value Representation

The Unknown (UN) VR shall only be used for Private Attribute Data Elements and Standard Data Elements previously encoded as some DICOM VR other than UN using the DICOM Default Transfer Syntax (Implicit VR Little Endian), and whose Value Representation is currently unknown, or whose known Value Representation is none of OB, OD, OF, OL, OW, SQ, UC, UR or UT and whose value length exceeds 65534 ($2^{16}-2$) and therefore cannot be encoded as a 16-bit unsigned integer in the Value Length Field defined for the known Value Representation (see ???). As long as the VR is unknown the Value Field is insensitive to byte ordering and shall not be 'byte-swapped' (see ???). In the case of undefined length sequences, the value shall remain in implicit VR form. See ??? for a description of Private Data Attribute Elements and section 10 and ??? 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 ???).

The UN VR shall not be used for File Meta Information Data Elements (any Tag (0002,xxxx), see ???).

Note

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

1 default, and not having been translated to the DICOM Default Transfer Syntax default in the interim, will have a known
2 VR.

- 3 2. If at some point an application knows the actual VR for an Attribute of VR UN (e.g., has its own applicable data dictionary),
4 it can assume that the Value Field of the Attribute is encoded in Little Endian byte ordering with implicit VR encoding,
5 irrespective of the current Transfer Syntax.
- 6 3. This VR of UN is needed when an explicit VR must be given to a Data Element whose Value Representation is unknown
7 (e.g., store and forward).
- 8 4. This VR of UN is also needed for the encoding of Data Elements with explicit VR whose value length exceeds 65534
9 (2¹⁶-2) (FFFEH, the largest even length unsigned 16 bit number) but which are defined to have a 16 bit explicit VR
10 length field.
- 11 5. The length field of the Value Representation of UN may contain the value of Undefined Length, in which case the contents
12 can be assumed to be encoded with implicit VR. See ??? to determine how to parse Data Elements with an Undefined
13 Length.
- 14 6. An example of a Standard Data Element using a UN VR is a Type 3 or Type U Standard Attribute added to an SOP
15 Class definition. An existing application that does not support that new Attribute (and encounters it) could convert the
16 VR to UN.
17

18 10.1 DICOM Default Transfer Syntax

19 DICOM defines a default Transfer Syntax, the DICOM Implicit VR Little Endian Transfer Syntax (UID = "1.2.840.10008.1.2"), which
20 shall be supported by every conformant DICOM Implementation. This implies that:

- 21 a. If an Application Entity issues an A-ASSOCIATE request, it shall offer the DICOM Implicit VR Little Endian Transfer Syntax in at
22 least one of the Presentation Contexts associated with each offered Abstract Syntax.

23 Note

24 Offering Abstract Syntax (AS1) in two Presentation Contexts with Transfer Syntaxes (TS1) and (TS2) is not valid, but
25 offering AS1-TS1, AS1-TS2 and AS1-TSD is valid because the DICOM Default Little Endian Transfer Syntax (TSD) is
26 present in at least one of the Presentation Contexts that are based on Abstract Syntax (AS1).

- 27 b. If an Application Entity receives an A-ASSOCIATE indication corresponding to a request that follows the requirements specified
28 in Section 10.1 (a), every Presentation Context related to a given Abstract Syntax cannot be rejected in an A-ASSOCIATE response
29 for the reason that none of the Transfer Syntaxes are supported.

30 Both of these requirements, (a) and (b), are waived when the Application Entity sending the pixel data has only access to the pixel
31 data in lossy compressed form or the pixel data in a lossless compressed form that is of such length that it cannot be encoded in the
32 default Transfer Syntax, and a Transfer Syntax that uses a pixel data reference is not offered.

33 Requirement (b) to accept the default Transfer Syntax is waived if a Transfer Syntax that uses a pixel data reference is offered.

34 Note

35 In other words, every sending AE is required to be able to convert any Data Set it is going to transmit into the default
36 Transfer Syntax, regardless of the form in which it originally received or stored the Data Set, except in the cases of when
37 the decompressed Pixel Data is too large to encode in the default Transfer Syntax or is received in a lossy compressed
38 form. In the case of lossy compressed Pixel Data, the sending AE is permitted to propose only the lossy compressed
39 Transfer Syntax appropriate to the lossy form that was received. In the case of lossless compressed Pixel Data that is too
40 large to encode in the default Transfer Syntax, the sending AE is permitted to propose any appropriate lossless compression
41 Transfer Syntax, not necessarily that in which the image was received, as an alternative to the default Transfer Syntax.

42 This waiver does not apply to Data Sets received in a lossless compressed form if the decompressed Pixel Data is small
43 enough to encode in the default Transfer Syntax, which means that any AE receiving a Data Set in a lossless compressed
44 Transfer Syntax that needs to re-send the Data Set is required to be able to decompress it in order to support (at least) the
45 default Transfer Syntax.

46 **Similar concerns apply to the Web Services transactions and are addressed with specific requirements in PS3.18.**